

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
31 May 2001 (31.05.2001)

PCT

(10) International Publication Number  
WO 01/38976 A1

(51) International Patent Classification<sup>7</sup>: G06F 9/44

(21) International Application Number: PCT/US99/27883

(22) International Filing Date:  
24 November 1999 (24.11.1999)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: CAMELOT IS-2 INTERNATIONAL d.b.a  
SKYVA INTERNATIONAL [US/US]; 25 First Street,  
Cambridge, MA 02141 (US).

(72) Inventors: BEHRMANN, Anja; 77 Verndale Street,  
Brookline, MA 02146 (US). BHARGAVA, Rohit; 133  
Hampshire Road, Wellesley, MA 02481 (US). DE-  
BOECK, Yvan; Apt. 305, 289 Highland Avenue,  
Somerville, MA 02144 (US). DHILLON, Bhupinder, S.;  
Apt. 3R, 4 Watson Street, Cambridge, MA 02139 (US).  
FRALEIGH, Stephen, P.; 134 Crossbow Lane, North  
Andover, MA 01845 (US). GROS, Oliver; 54 Harvard

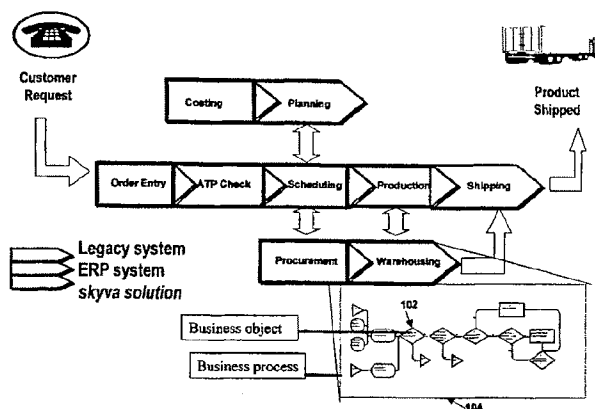
Street, Medford, MA 02155 (US). JONAS, Joerg; Apt.  
C, 107 Amory Street, Cambridge, MA 02139 (US). LIP-  
TON, Michael, J.; 64 Hill Street, Lexington, MA 02421  
(US). LJUNGBERG, Magnus; 235 Lincoln Street, Lex-  
ington, MA 02421 (US). MATTHEWS, Brian, L.; 302  
Boston Road, Groton, MA 01450 (US). MCGILLIVRAY,  
Duncan, N.; 34 Chatham Street, Cambridge, MA 02139  
(US). MOECKESCH, Guenther; Apt. 609, 1008 Mass  
Avenue, Cambridge, MA 02138 (US). O'DONNELL,  
Timothy, J.; 3 Rio Vista Street, Billerica, MA 01862 (US).  
PERRY, Alan, Charles; 49 Saw Mill Road, Concord,  
MA 01742 (US). TEICHMANN, Lutz; Kurfuerstenallee  
14, 69181 Leimen (DE). VAN KHUIJKELOM, Hans;  
Apt. 3, 47 Cypress Street, Brookline, MA 02446 (US).  
ZILLI, Kaj, O.; 5108 Stearns Hill Road, Waltham, MA  
02451 (US). ZINK, Andreas, W.; 15 Plymouth Street,  
Arlington, MA 02174 (US).

(74) Agent: RZUCIDLO, Eugene, C.; Graham & James LLP,  
885 Third Avenue, New York, NY 10022 (US).

(81) Designated States (*national*): AL, AM, AT, AU, AZ, BA,  
BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES,  
FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP,

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR BUSINESS MODELING



(57) Abstract: A method and apparatus that define in a graphical way how to generate an application or template or custom solution based on a user defined Business Process Model, without programming. The computerized system uses prefabricated software components. These components are described in the Solution Library. The Solution Library includes a list of exclusive and completely exhaustive components (Business Patterns, data mapping models, state models). These components build up generic solutions for all business applications across the industry. The prefabricated software components include order elements, resource elements and process elements, with a state table attached. They are composed into Business Patterns. The present invention provides an analysis of an optimal business process. Once the optimal business process is defined, a methodic mapping of preexisting computer software systems/third party software systems to the business process can demonstrate functionality gaps. A gap exists if preexisting computer software system/third party system is unable to meet with the functional requirements dictated out of the business process. These gaps will be filled with components out of the Solution Library by: extending the Solution Library components to fulfil the specific industry and customer needs, through configuration, without programming, and by integration of existing solutions. Once the gaps are filled, the computer software system is operational and spans the user's entire business process.

WO 01/38976 A1



KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW.

**Published:**

— With international search report.

(84) **Designated States (regional):** ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

## Method and Apparatus for Business Modeling

### BACKGROUND OF THE INVENTION

The present invention relates generally to computer programs and, specifically, to a method and apparatus for allowing a person to implement business computer software  
5 without having to write computer programs.

Today, competitive advantage belongs to companies that benefit from flexible and dynamic business processes. Unfortunately, many of today's supply chain solutions are unable to support a flexible and dynamic environment. The most popular software packages force customers to fit their business to the software. This requirement causes incredibly high  
10 implementation costs, upgrade costs, and clumsy modification of the customer's business. In other words, a customer's business is limited by the software.

To be flexible and dynamic, customers need something quite different. They need standard software that allows them to: easily shape the software to fit the specific needs of their company (not vice versa) and dynamically change the way they do business (on the fly)  
15 independent of software releases. Customers also need to successfully manage business processes across companies (partners) and systems, and to integrate the many systems with which they work to achieve supply chain success. The current generation of supply chain software simply cannot meet these needs.

In addition, companies today need to model supply chain processes for  
20 implementation, monitoring, and improvement; share supply chain processes and data across companies; model employees' work environments to meet their specific tasks and needs (that is, they need a scalable solution); generate supply chain solutions from models of their company-specific business processes; manage business processes across multiple software systems; be dynamic (in other words, be able to change or react very quickly); and have a  
25 supply chain solution that is standard, and yet, fits their company.

The following sections explain these requirements in more detail. For example, one large computer hardware manufacturer has a very specific way of handling direct ordering. Their processes, however, are most likely different from processes at other companies in their industry. They need the freedom to run their business the way they want to and software that  
30 will support that freedom. Furthermore, individual company philosophies like "order today,

get it tomorrow" demand very specific ordering and delivery processes that software should support, but not force on *every* company.

Companies also to be able to run *what ifs* to see the effects of changes or processes without implementing the changes. With *what ifs*, companies can inexpensively optimize the way they do business. Companies need to be able to concentrate on their business, not on the software that supports that business. This means that software must be transparent and efficiently allow them to run their businesses. Any software solution short of this requirement is unacceptable.

Companies need to share supply chain processes and data. Many companies have learned that one method of gaining competitive advantage is to partner with other companies that can handle specific tasks better than they themselves can. This means that companies today are open to partnership. This also means that business processes cross company lines. Crossing company lines can mean outsourcing processes, using an extended supply chain, or any number of additional possibilities.

If business processes cross company lines, then data needs to be able to cross system lines as well. This means that any software solution today needs to be open to working with numerous technologies (for example, Internet, EDI, BAPIs). Partners or customers may not use the same systems and yet, need to be able to work with the data.

Companies need to customize employees' work environments. Even when many companies have business processes that are somewhat similar, the way the individual employees carry out those processes are probably quite different. For example, in a large company, one employee may enter sales orders, another might perform ATP checks, and yet another may create invoices. In a smaller company, however, one person may perform all of the above tasks. The employee at the larger company should not be forced to wade through unneeded data on the screen or series of screens. Similarly, the employee at the smaller company should not be forced to call three separate transactions.

Companies should have the opportunity to define the specific environments with which their employees work, thereby giving employees just what they need to be successful at their job. With such an environment, they can eliminate confusion, increase productivity, and reduce the chances of error. And, of course, companies should be able to do all of this without extreme pain or coding.

Companies need to be able to generate supply chain solutions from business models. If a company goes through the great effort of modeling its business processes down to the

smallest details it would make the most sense if, from that model, they could automatically generate parts of its software solution. There is no reason that a company should ignore a model that it has toiled painstakingly over just to have to rework (configure) that model in a different system.

- 5           Companies need to manage business processes across multiple software systems. One of the challenges with integration is that data structures between systems are often different.

          In general, a company faces two challenges in a distributed environment: controlling data distribution and controlling process distribution. If, for example, a large beverage manufacturer distributes recipes, distributes work plans, and distributes material data, some  
10 parts of the data are stored and maintained centrally, and other parts of the data are decentralized. For example, the main mixture for cola may be centrally maintained, but the production or warehousing data may be maintained in multiple locations. As another example, jeans manufacturer executes production orders at the local level, but plans in a central system at the aggregation level. Currently, it would not have a global view of how  
15 many jeans, in which sizes, are actually finished and in stock..

          Companies need to be dynamic. Only companies that can maintain flexible and dynamic business processes will hold a competitive advantage. Companies need to be able to change the way they do business at the drop of a hat, at the customer's whim, or in the face of unexpected challenges. If a business model changes, software has to be able to do the same.  
20 This is problematic if the software must be changed by a human being writing new code. Furthermore, a company should also be able to simulate the impacts of changes beforehand using *what if* scenarios. Coding such what-if scenarios takes time that companies often do not have and cannot afford.

          One good thing about standard software is that companies don't have to develop it  
25 themselves. Many software packages force their users to run business their way, or only allow users to customize or modify their software by digging deep in the code—where they never know exactly what is affected.

## SUMMARY OF THE INVENTION

          A method and apparatus that define in a graphical way how to generate an application  
30 or template or custom solution based on a user defined Business Process Model, without programming. The computerized system uses prefabricated software components. These components are described in the Solution Library. The Solution Library includes a list of

exclusive and completely exhaustive components (Business Patterns, data mapping models, state models). These components build up generic solutions for all business applications across the industry. The prefabricated software components include order elements, resource elements and process elements, with a state table attached. They are composed into Business  
5 Patterns. Due to their generic nature, these patterns can be applied to various industry domains from process industries to service business. An example would be the business pattern, split order. A sales order is split into a manufacturing order and a purchase order. In this process, the physical orders are represented by the "abstract" order objects. The resource element used can be the person doing this order split, or the system doing the split. The  
10 process element contains the rules according to which the split is done.

The Solution Library is aimed at facilitating the development and implementation of distributed business applications for industries. Business applications include for example: available to promise, distributed master data, and integrated manufacturing execution for industries such as, automotive, process industries, services, etc.

15 The present invention provides an analysis of an optimal business process. Once the optimal business process is defined, a methodic mapping of preexisting computer software systems/ third party software systems to the business process can demonstrate functionality gaps. A gap exists if the preexisting computer software system /third party system is unable to meet with the functional requirements dictated out of the business process. These gaps  
20 will be filled with components out of the Solution Library by : extending the Solution Library components to fulfil the specific industry and customer needs, through configuration, without programming, and by integration of existing solutions. Once the gaps are filled, the computer software system is operational and spans the user's entire business process.

In accordance with the purpose of the invention, as embodied and broadly described  
25 herein, the invention relates to a method performed by a data processing system that allows a user to automatically generate a software computer system tailored to a particular business model, without programming by the user, comprising the steps of: facilitating the modeling of one or more business processes, business objects and business rules of the user to yield a business model having one or more business processes; optimizing the business model; if a  
30 business process does not correspond to a pre-existing computer software system, generating a new computer software system for the business process using a predefined software component library; and if a business process corresponds to pre-existing computer software

system, then generating integration code from new computer software systems into the pre-existing software systems using predefined integration software components.

5 In accordance with the purpose of the invention, as embodied and broadly described herein, the invention relates to a computerized method, comprising: providing mutually exclusive and completely exhaustive (MECE) software components for generating computer software systems for all business application across the industries, and providing software components for generating integration code into pre-existing software systems for commonly used pre-existing packaged software systems

10 Advantages of the invention will be set forth in part in the description which follows and in part will be obvious from the description or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims and equivalents.

#### BRIEF DESCRIPTION OF THE DRAWINGS

15 The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is a block diagram showing a model of a business process constructed with business objects and business rules specific to a customer or an industry.

20 Fig. 2 is a block diagram showing what parts of the business process are not covered by a preexisting software system.

Fig. 3 is a block diagram showing a Business Process Modeling and Simulation environment, a Business Object Configuration environment, Cross software systems Integration and data mapping environment and a User Interface Building environment.

25 Fig. 4 is a block diagram showing the mapping of data models/data structures of the new computer software system and the preexisting software system using the business objects designed in the business process.

Fig. 5 is a flow chart showing the major steps taken in a preferred embodiment of the present invention.

30 Fig. 6 is a diagram showing an action and how it relates to a business process.

Fig. 7 is a diagram showing a definition of a "to-be" model.

Fig. 8 is a diagram showing an example business process.

Fig. 9 is a diagram showing an example of a computer software system selected against requirements dictated by a business process.

5 Fig. 10 is a diagram showing the elements of a detailed modeling step.

Fig. 11 is a diagram showing dependencies between various system elements.

Fig. 12 lists exemplary attributes of the object facility.

Fig. 13 shows an exemplary state model for an order.

10 Fig. 14 shows an example of a different view of how the order goes from its FROM state to its TO state.

Fig. 15 shows a user interface screen, listing customer demand orders and proposed production to meet part of the order.

Fig. 16 shows an example of the pre-defined attributes of an order.

15 Fig. 17 is a table showing an extension of the order example, with some attributes and justification aspects for the order.

Fig. 18 shows an example of how extension of the data can be driven by some industry specifics.

Fig. 19 shows the overall structure of the various elements covered under the software components, and how they are grouped in a hierarchical order.

20 Fig. 20 is an overall view of the business process template composition.

Fig. 21(a) is a block diagram showing an exemplary data processing system implementing an embodiment of the present invention.

Fig. 22 shows an example of data mapping.



Fig. 23 is an exemplary screen dump of a screen that facilitates the user's designing a business object.

## DETAILED DESCRIPTION OF EMBODIMENTS

Reference will now be made in detail to several embodiments of the present invention, examples of which are illustrated in the accompanying drawings. Wherever practicable, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

### A. General Discussion

The present invention allows a business person or an inexperienced programmer to model a business process and design software for that business process. The following discussion relates to the general environment in which software in accordance with the present invention operates. Later discussion relates to specific examples of the present invention.

Fig. 21(a) is a block diagram showing an exemplary data processing system 2100 implementing an embodiment of the present invention. Data processing system 2100 communicates with the others via a network 2114. Network 2114 can be the internet, a WAN, a LAN, a wireless network, a cellular telephone network, a radio frequency network, or any other appropriate network on connection. System 2100 includes at least one processor 2102 and storage 2104 (such as a memory). Storage 2104 of system 2100 includes a variety of software and data structures, including business pattern(s) 2150, business process template(s) 2152, industry template(s) 2154, integration template(s) 2156, and at least one customer template 2158. Storage 2104 also includes software required to implement the methods and data structures described herein. The steps of the described embodiment of the present invention are performed when instructions of a computer program stored in storage 2104 (or similar storage) are performed by processor 2102 (or another appropriate processor).

System 2100 also includes at least one input device 2118, such as a keyboard, touchscreen, mouse, or the like. System 2100 also includes at least one output device 2120 such as a printer, display screen, or the like. System 2100 also includes a computer readable medium input device 2122 and a computer readable medium 2124. Computer readable medium 2124 can be any appropriate medium that has instructions and/or data structures

and/or objects stored thereon. These instructions etc. are loaded from computer readable medium 2124 into storage area 2104. Instructions etc. can also be loaded into storage area 2104 in the form of a carrier wave or similar signal over network connection 2114. Thus, the instructions and data in storage 2104 can be loaded into storage via an input device, via a network, such as the internet, a LAN, or a WAN, or can be loaded from a computer readable medium such as a floppy disk, CD ROM, or other appropriate computer readable medium. The instructions can also be downloaded in the form of a carrier wave over a network connection.

System 2100 also includes an operating system (not shown). A person of ordinary skill in the art will understand that the storage/memory also contains additional information, such as application programs, operating systems, data, etc., which are not shown in the Figure for the sake of clarity. It also will be understood that data processing system 2100 (or any other data processing system described herein) can also include numerous elements not shown, such as additional data, software, and/or information in memory, disk drives, keyboards, display devices, network connections, additional memory, additional CPUs, LANs, input/output lines, etc.

It will be understood that the invention is not limited to any particular implementation or programming technique and that the invention may be implemented using any appropriate techniques for implementing the functionality described herein. The invention may be implemented in any appropriate operating system using any appropriate programming language or programming techniques and may be implemented as hardware, software, or any appropriate combination of hardware and/or software.

#### B. More Specific Discussion

A computerized system is provided which defines in a graphical way how to generate an application or template or custom solution based on a user defined Business Process Model, without programming. This system is called the Solution Environment (a preferred embodiment is called the "Skva Solution Environment"). The computerized system comprises the mapping of data models/data structures of the new computer software system and the pre-existing computer software system using the business objects designed in the business process (Figure 4) uses prefabricated software components in a methodic way. These components are described in the Solution Library. (Pre-existing computer software is

sometimes called third party computer software, since some pre-existing software is available from third parties.)

The Solution Library includes a list of exclusive and completely exhaustive components (Business Patterns, data mapping models, state models). These components  
5 build up generic solutions for all business applications across the industry. Figure 22 shows an example of data mapping in which Enterprise Resource Planning (ERP) data and scheduling data are mapped to an object framework.

The prefabricated software components include order elements, resource elements and process elements, with a state table attached. They are composed into Business Patterns.  
10 Due to their generic nature, these patterns can be applied to various industry domains from process industries to service business. An example would be the business pattern, split order. A sales order is split into a manufacturing order and a purchase order. In this process, the physical orders are represented by the "abstract" order objects. The resource element used can be the person doing this order split, or the system doing the split. The process element  
15 contains the rules according to which the split is done.

The Solution Library is aimed at facilitating the development and implementation of distributed business applications for industries. Business applications include for example: available to promise, distributed master data, and integrated manufacturing execution for industries such as, automotive, process industries, services, etc.

20 The composition of the solution is achieved by using the Solution Environment. This environment is a methodic analysis of the optimal business process. Once the optimal business process is defined, a methodic mapping of preexisting computer software systems software to the business process can demonstrate functionality gaps. A gap exists if the preexisting computer software system is unable to meet with the functional requirements  
25 dictated out of the business process. These gaps will be filled with components out of the Solution Library by extending the Solution Library components to fulfil the specific industry and customer needs, through configuration, without programming and by integration of existing solutions

Once the gaps are filled, the computer software system is operational, it now spans the  
30 entire business process.

Different industries seem to have specific requirements for specific business applications. Across the industries, we have found a list of **completely exhaustive and mutually exclusive common Business Patterns and a single data model**. These can be

extended to meet the specifics in every industry and business application. Because the business patterns are exhaustive, they cover every possible business. Because they are exclusive, only one is appropriate in a given situation.

5 The solution environment comes with a unique **methodology and set of unique tools** that enable the building of a business application based on the customer's business process model. The business process model includes definition of business hierarchies, business objects and business rules.

A solution is composed using business patterns. This composition is carried out with a unique methodology which includes the following steps:

- 10
- Model the business process, business objects and business rules of a customer or industry. This model is the graphical representation of the way the end user executes his business process.
  - Optimizing the business process model
  - A solution gap analysis defines which parts of the business process are covered by

15

  - preexisting software systems, and which parts are not covered.
  - if the business processes are not found in a preexisting computer software system, generates the new computer software system from the business process using a predefined software component library
  - if the business process is covered in a preexisting computer software system, then

20

  - automatically generating the integration code from the new computer software system into the preexisting computer software system using predefined integration software components.

25 The Solution Environment, used in the methodology, provides the environment for the generation and configuration of the business application without the need for programming. The environment includes:

- Business Process Modeling and optimizing
- Business Object Configuration
- Cross preexisting software systems integration and data mapping
- Graphical user interface building

30

The methodology makes further use of Business Patterns and Business Process Templates as building blocks to fill the functionality gaps.

A graphical representation of the composition of computer software system is given through Figures 1 to 4. The methodic generation of the customer software solution starts

with a graphical representation of the business process. The model is constructed with business objects (e.g., business object 102) and business rules (e.g., business rules 104) specific to a customer or an industry (Figure 1).

5 In the business process 202, map the different business processes to software systems, and find out what parts of the business process are not covered 204 by a preexisting software/third party software system (Figure 2).

The functional gaps that will be discovered on Figure 2 must be filled with a software system. In order to fill the functionality gaps we offer business process templates, industry templates and integration templates. These are assembled and configured using the Solution  
10 Environment 302. This environment includes a Business Process Modeling and Simulation environment, a Business Object Configuration environment, Cross software systems Integration and data mapping environment and a User Interface Building environment (Figure 3).

A final step is the generation of the integration code from the new computer software  
15 systems into the third party software systems using predefined integration software components. It comprises mapping of data models/data structures of the new computer software system 402 and the third party computer software system 404 using the business objects designed in the business process (Figure 4).

## 1 The Solution Environment

20 The Solution Environment contains the tools to compose the computer software system. A unique methodology is used to make these tools work. This method begins with the business process definition, and by making use of the Solution Environment, generates the computer software system. The methodology is graphically represented in Figure 5 to 11. Figure 5 details the major steps taken in the methodology. Each of the blocks is further  
25 detailed in a following paragraph. The legend for looking at the following paragraphs is given in Figure 6.

### 1.1 Model the 'to-be' Process

Further detailing the step described above is the definition of the "to-be" model, Figure 7. When the to-be model is defined, a simulation of the model will lead to the  
30 optimal solution, optimizing this business process model. One will define a list of business opportunities, define alternative scenarios based on these opportunities, simulate the scenario and validate to see if this is a suitable solution. Once validated and found optimal, this

solution found will serve as the input into the next step, which system is to be selected to fit this Business Process.

An example business process is shown in Figure 8. In this example an order comes in and is processed. One level under order processing is the process of checking the order credit. The example shows various elements of the model like the business processes and actions, the events, and resources.

## 1.2 System Selection

In order to map a business process into a computer software system, the following procedure is followed.

10 In the different business processes of Figure 9 a computer software system is selected against requirements dictated by the business process. If the business processes are not found in a preexisting computer software system, generates the new computer software system from the business process using a predefined software component library.

15 If the business process is covered in a preexisting computer software system/third party software system, then automatically generates the integration code from the new computer software system into the third party computer software system using predefined integration software components

## 1.3 Detailed Modeling

20 The detailed modeling is the transition from the business process, obtained through the process of in Figure 7, towards defining what elements of the Solution Library need to be applied in order to compose, without programming, the custom software computer system. Figure 10 describes the elements of the detailed modeling step.

Different terms are used in Figure 10. In the following we will define these terms. For convenience the terms are marked bold, such will bring them forward in the text.

## 25 Business Process and Business Object

Business Processes may describe business activities performed by the user, or by the computer software system. A business process is made from a combination of other business processes. A Business Object is a data structure to store dynamic information as an output of a business action. The information contained within a business object, an **attribute**, will be

used as the input for business rules and business actions . A business action is an action performed on a business object, which may change the state associated with that object. The business rule defines a decision pattern that needs to be followed.

These different elements are displayed in Figure 11, which also highlights their dependency. **Attributes** are the basic construct of the data structure definition. An attribute defines an abstract data type. For example, Figure 12 shows some attributes to the object facility: its identifier, name and a deleted or not description. Each data structure is composed of a number of optional logical groups of attributes called **aspects**.

The **state model** defines the dynamic behavior of the business object in terms of a set of states that the business object goes through during its life span. An external event – either sent by the user or by the system – to the business object, triggers a transition defined in the state model associated with the target business object. The transition that is triggered depends on the state that the business object is in, at the time the event is delivered. If the trigger rule conditions (associated with the transition) evaluate to true, the actions associated with the transition are executed and push the business object along its life-cycle. Each transition associated with the state of the business object is evaluated in turn until one is taken (conditions evaluate to true) or if no transition is taken, the event is discarded.

Figure 13 shows a state model for an order. A different view is how the order goes from its FROM state to its TO state. See Figure 14. DMO is the Detailed Manufacturing Order.

The **User Interface** is a finally defined with its user interactive screens. Figure 15 shows a user interface screen, listing customer demand orders and proposed production to meet part of the order.

## 4 SOFTWARE COMPONENTS

### 4.1 Abstract

The Solution Library contains a set of software components. The software components are **resource elements, order elements, and process elements**, and explained in detail hereafter. A business application library is a pre-configured assembly of business patterns and business objects to form business process templates, industry templates, integration templates. The library can be oriented towards industry-neutral demand and supply-related problems, such as:

- Production order planning,
- Production order scheduling
- ATP check
- Maintenance scheduling,
- 5 · Production order execution and confirmation.

The library contains the following:

- Base objects and methods organized into **software components**.
- Reusable domain-level business process software components:

### Business Patterns

- 10 · Groups of business patterns assembled into **industry templates, business process templates, and integration templates**

### 4.2 Software components definition

- The software components are the business objects in the Solution Library. They include **resource elements, order elements, and process elements** and are related to each other in various ways. Those relationship defines the object hierarchy. These three objects are very abstract in the way they represent business objects and business processes of an enterprise.

The software components must be:

- Independent - meaning that the configuration of one component does not affect the configuration of others
- 20 - Mutually exclusive - meaning that there is no conceptual overlap between software components. Thus, there is exactly one object in the Solution Library responsible for storing a particular piece of information or carrying out a specific function.
- Completely Exhaustive - meaning that there is always one part in the Solution library responsible for all anticipated kinds of objects and business activities.
- 25

#### 1.1.1 Software components list

- Resource elements** are the means to execute business activities, e.g. work centers, tools, raw materials, finished products, and waste. Resource elements may be grouped. There are various criteria for grouping resource elements, e.g. product groups and organizational units.



**Order elements** are used as software components for decision communication and tracking in a real world where communicating partners are separated by aggregation and location differences in their view of the supply chain.

A **process element** is used to describe how one may carry out a business activity.

- 5 Relative and/or conditional timing of the (sub) business activities involved is an important part of this description. The business rules define how the process is executed, according to what rules.

As an example, a sales order comes in and is split into a manufacturing order and a purchase order. The two kinds of physical orders (manufacturing and purchase) are  
10 represented by abstract order objects, the department where the order splitting takes place is the resource, and the process element would hold information on how to split the order, according to business rules. In this example, the resource used is the work center to process and order through a process element.

#### 1.0.0 Predefined aspects and attributes

- 15 Every software component can have various **predefined aspects** and **attributes**. An attribute is data field, which contains data belonging to that object. Attributes can be grouped in a logical order to form aspects. For example an order has these aspects predefined to it.

##### Order justification :

The justification part of an order describes the reason for the existence of the order.

- 20 The justification section includes references to other orders that caused this order to be instantiated. It also contains the due date, the product and the quantity to be produced.

##### Order decision :

- 25 The decision part of an order element collects the decisions associated with the processing of an order. These decisions may include references to the predicted process that is linked to the order and decisions concerning the predicted due time of the order.

##### Order progress:

- 30 The progress part of an order element is used for keeping the tracking information that is reported from the fulfillment of the order. This information is propagated from one order to the superior order using the individual order justifications.

An example of the pre-defined attributes of an order are given here and described in Figure 16. A sales order leads into a purchase order and an aggregate manufacturing order. So the order justification part would contain the information with respect to the order network that exist, as is the case in Figure 16. The order net is the network of orders that is generated based on then sales order, it is the dis-aggregation of an order into other orders.

The decision aspect of the sales order could contain information such as when this order needs to be delivered, or a reference to the process to be followed to fulfill the order on the manufacturing order.

The progress aspect holds the data related to the actual progress of the order such as the total quantity produced at a given time on the manufacturing order.

### 1.0.0 Aspects and attributes for a particular solution

On a purchase order for example, one will define specific attributes for a specific computer system. The order example has been extended in the table in Figure 17, with some attributes and **justification aspects** for the order.

Extension of the data can be driven by some industry specifics as described in Figure 18.

## 1.0 Business Patterns

Business patterns complement the software components by providing generic functions, systems integration functions and industry specific functionality in a distributed environment. Figure 19 shows the overall structure of the various elements covered under the software components, and how they are grouped in a hierarchical order. The Figure covers the business patterns, its components, and how business patterns build up business process templates.

Business patterns may be constructed from sub-business patterns. They interact with objects in the repository, with users, and with agents. The interfaces for these interactions are defined as roles within the business patterns. To execute the business pattern, the scripting language provides information that identifies the objects, the users and the agents that will play the roles defined in the business pattern. Communication between the roles and the players takes place using **role adapters**, which allow a 'plug and play' approach to business pattern use. This increases the reusability of the business pattern.

Conditions are also used to affect the behavior of the business pattern. These may be associated with the business pattern itself, the objects, the users or the agents associated with

the roles of the business pattern. Typically, these conditions implement business rules or provide an algorithm that is specific to the context in which the business pattern is used.

**Role adapters** are used to adapt the interface on an object for a specific requirement. The types of adaptation that a role adapter can provide include:

- 5        Event transformation - when events generated by an object cannot be recognized by a recipient.

Terminology transformation - when a generic interface on an object must be presented according to the expectations of the problem domain (e.g. a resource element is presented as a work center).

- 10       Data transformation - when generic data is presented in a specific way to meet the needs of the data consumer (e.g. a user needs to see text, numeric values, currency values etc. according to a specific language and country requirements).

For example, a business pattern providing a generic order generation capability may be used in a wide variety of application contexts, such as planning, production, or warehouse management. This business pattern can be customized for each application through conditions that govern how the pattern will behave in the specific context. Order generation may involve the use of a 'create order' subpattern whose responsibility it is to create a new order representation and manage order ID generation. 'Create order' will use the order ID generation condition to allow customization of how order IDs are generated.

## 20    **4.4    Business Process Templates**

A Business Process template is a set of business patterns that implement a set of recognized business practices. A business process template might be associated to a specific industry, it then becomes an industry specific business process template. Business process templates may also serve as integration enablers of software systems, than they are integration derived business process templates. Figure 20 is an overall view of the business process template composition. In a specific customer case the computer software system is composed out of extended business process templates. Figure 23 is an exemplary screen shot of a process allowing the user to design a business object.

## 5       **General Terminology**

### 30    **Scripting Language (SSL)**

SSL is a general-purpose scripting language. It is used for the following purposes:

- To model a Business Processes,
- To describe an agent behavior, and
- To define a descriptive Business Processes.

### **Chain of Responsibility**

- 5 Chain of responsibility describes the behavior of a system that, if an object is unable to provide a given function or service, may pass the requirement to some other object. This object, in turn, may either handle the requirement or pass it on to another object.

This technique is very powerful when used in association with a hierarchy of objects, where a generic technique may be implemented at the root of the hierarchy and more specific

- 10 techniques optionally implemented as the hierarchy expands and specializes.

Business Patterns may be organized within chain-of-responsibility relationships. This allows them to be looked up according to a hierarchy, thereby reducing the need for a user to define policies for every single object in the system.

### **Control Systems Wrapper**

- 15 Control Systems Wrappers must interface with any number of external control systems. Often, these systems communicate using their own proprietary mechanisms. To be able to communicate with them in a standard way, has defined a standard interface, called a wrapper that can take communication from one system and transpose it to the other system. Such interfaces are part of a particular solution (not the templates). However, the standard
- 20 template defines a standard format that an interface to an external system must follow.

### **Event Process**

- The objective of Business Process modeling is to derive event processes. Uses event processes to drive an object (such as an order) through the business activities defined in the Business Process. Event processes are represented as a state model for the object, and are
- 25 realized by scripting together Business Patterns using Scripting Language (SSL).

### **Role Adapter**

Role adapters are used to adapt the interface on an object for a specific requirement. The types of adaptation that a role adapter can provide include:

Event transformation - when events generated by an object cannot be recognized by

a recipient.

- Terminology transformation - when a generic interface on an object must be presented according to the expectations of the problem domain (e.g. a resource element is presented as a work center).
- 5 · Data transformation - when generic data is presented in a specific way to meet the needs of the data consumer (e.g. a user needs to see text, numeric values, currency values etc. according to a specific language and country requirements).

### **Scenario**

- 10 A scenario is a configuration device for configuring the operational behavior of deciding agents in an operational context. Scenarios are stored as predefined sets of configuration data. The right scenario in a particular context can be selected either automatically or by a human agent. Editing can only be done by a human agent.

### **Templates**

#### **Control**

- 15 Controls, sometimes referred to as 'widgets', are the basic elements from which panels and screens are built.

#### **EXAMPLE:**

Type-in boxes, list boxes, scroll bars, etc.

#### **Business process template**

- 20 A business process template is a set of Business Patterns. The industry business process template would in many cases be built upon an industry-specific information. The following shows how a custom computer software system is built from software components, business patterns, and an industry business process template.

- demand-supply assignment – low-level method of a software component
- 25 · order creation – business pattern
- production order creation – part of an industry template
- production order creation with a specific rule or condition – part of a custom computer software system

### **Identifying Attribute**

An identifying attribute is an attribute of an object that uniquely identifies the object that is used by an end user to identify the target object for some user interface function.

#### **EXAMPLE:**

- 5 An order ID is an identifying attribute for an order.

### **Non-Identifying Attribute**

- A non-identifying attribute is an attribute of an object that is a non-unique key that helps identify an object. It is used by an end-user to identify the target object for some user interface function. Typically, these non-unique keys will result in a list of objects from which the user can select the target.
- 10

#### **EXAMPLE:**

A customer ID is a non-identifying attribute for an order.

### **Panel**

- Panels are used to build screens. They consist of groups of interface software components called controls, which perform common tasks. Again, exploits commonality by providing a set of standard panels, which can be configured and used in many screens.
- 15

#### **EXAMPLE:**

A tree-view panel could be used to view a hierarchy of objects, which could be a process description, or an order net.

### **20 Screen**

- A screen is a high-level component that performs major elements of functionality in . Even though they operate in different functional contexts, screens often perform standard functions, like selecting objects, viewing hierarchies, and editing objects. Exploits this similarity by providing a library of standard screens which are parameterized, so that in template configuration, they can be adapted to different object classes with different rules.
- 25

## **Modeling**

### **Business Event**

A business event is a message generated by the system occurring as a result of a business trigger. Other Business Processes may be launched by or waiting for the business event

### 5 **Business Object**

A business object is an object in the database that is an input or an output of a Business Process. However, it is not necessary to identify all business objects, just the 'central' ones.

#### **EXAMPLE:**

Production order and sales order.

### 10 **Business Policy**

A business policy is a business step that describes in what way the execution of the same Business Process differs in different business units. It is a mechanism that allows the modeler to abstract Business Processes for different business units into a single model (e.g. orders are always split in Plant 1 but they are never split in Plant 2). A business policy may  
15 also be used to qualify the same referenced Business Process.

### **Business Process**

A Business Process is a step in either a business scenario or in a higher-level Business Process. Business Processes may describe business activities performed by the user (user process), or by the system (system process). A Business Process has a combination of other  
20 Business Processes.

### **Business Process Modeling**

The description of a Business Process, typically tracing the flow of business documents, such as orders, in terms of the individual steps performed, and the alternative paths that the document may take. Often, there is an 'as-is' model, covering the existing process, and a 'to-be' model, covering the newly designed or reengineered process.  
25

**Business Process Simulation**

The use of computerized simulation tools to simulate a Business Process model in order to verify its accuracy, to measure various business metrics such as response times, resource utilization, and to perform what-if analysis.

**5 Business Rule**

A business rule is a business step that is a pause or a branch in the flow, and which is dependent upon something (a condition). This condition can be based on the state of a business object (e.g., the order is valid), or on the occurrence of a business event (e.g., an ATP result has arrived via workflow).

**10 Business Scenario**

A business scenario is a set of Business Processes to fulfill a particular purpose in a business unit. A set of business scenarios describes all of the business activities of this business unit. A business scenario contains at least one Business Process.

**Business Step**

**15** A business step is used to complement a Business Process. There are two kinds of business steps:

- Business triggers (including workflow notifications)
- Business rules (including waits)

**Business Trigger**

**20** A business trigger is a business step that launches another Business Process, or generates a business event that causes another Business Process to do something.

**Business Unit**

A business unit is a unit of a business hierarchy. Business units are organizations, physical locations, resources, or products.

**25 Unit of Work**



The unit of work is a Business Process at the level of detail where, according to the methodology, the modeler stops drilling down. Technically, this corresponds to the level at which functionality can be performed by Business Patterns. Functionally, this is determined according to the following criteria:

- 5     · The level of a screen or panel (subscreen) has been reached
- Single logical business activity
- Activity that is performed by one user (or the system automatically) at one point in time
- Must be the same size or smaller than a transaction (where a user would say OK or
- 10     Cancel)
- The lowest level of user involvement ('what' not 'how')
- The level where the system does not give the user a choice

### Use Case

- A use case is a particular situation that necessitates the slight variation of attributes to objects to address differences in business processes.

### EXAMPLE:

In the order entry use area, there may be separate use cases for new orders, reorders, EDI orders, changes to orders, etc.

### State Model

- 20 A state model is a description of the program an agent will follow. It is a mechanism the system builder uses to define the objects' specific steps and actions.

It includes the following software components:

- The specific fixed or optional steps (called states) in an object's life cycle,
- The specific events that can be expected when an order object is in a specific state,
- 25 · Any specific conditions which must be fulfilled before certain actions are taken,
- The specific actions that the computer software system must perform when the conditions are fulfilled. One very important action that must be specified is what the new state of the object will be after the actions are completed, and

The exceptions that may arise when executing the actions and what is the new state. However, not every type of agent behavior can be described by a state model. State models are only used in special types of agents, i.e. agents that track or guide objects through their state life cycle after they have been created by other agents.

5 In summary, the present invention allows a user to define in a graphical way how to generate an application or template or custom solution based on a user defined Business Process Model, without programming. The computerized system uses prefabricated software components. These components are described in the Solution Library. The Solution Library includes a list of exclusive and completely exhaustive components (Business Patterns, data  
10 mapping models, state models). These components build up generic solutions for all business applications across the industry. The prefabricated software components include order elements, resource elements and process elements, with a state table attached. They are composed into Business Patterns. Due to their generic nature, these patterns can be applied to various industry domains from process industries to service business.

15 The Solution Library is aimed at facilitating the development and implementation of distributed business applications for industries. Business applications include for example: available to promise, distributed master data, and integrated manufacturing execution for industries such as, automotive, process industries, services, etc.

The present invention provides an analysis of an optimal business process. Once the  
20 optimal business process is defined, a methodic mapping of preexisting computer software systems/third party software systems to the business process can demonstrate functionality gaps. A gap exists if the preexisting computer software system /third party system is unable to meet with the functional requirements dictated out of the business process. These gaps will be filled with components out of the Solution Library by : extending the Solution Library  
25 components to fulfil the specific industry and customer needs, through configuration, without programming, and by integration of existing solutions. Once the gaps are filled, the computer software system is operational and spans the user's entire business process. Accordingly, it is intended to embrace all such alternatives, modifications and variations as fall within the spirit and scope of the appended claims and equivalents.

## WHAT IS CLAIMED IS:

1. A method performed by a data processing system that allows a user to automatically generate a software computer system tailored to a particular business model, without programming by the user, comprising the steps of:
  - 5                   facilitating the modeling of one or more business processes, business objects and business rules of the user to yield a business model having one or more business processes;  
  
                    optimizing the business model;  
  
                    if a business process does not correspond to a pre-existing computer software  
10           system, generating a new computer software system for the business process using a predefined software component library; and  
  
                    if a business process corresponds to pre-existing computer software system, then generating integration code from new computer software systems into the pre-existing software systems using predefined integration software components.
- 15           2. The method according to claim 1 wherein said facilitating the modeling of the business process includes facilitating designing of business processes, business objects, business hierarchies and business rules.
3. The method according to claim 1 wherein optimizing the business model includes:
  - 20                   simulating business process alternatives, and  
  
                    identifying the optimal business process in terms of improving the companies key success factors.
4. The method according to claim 1 wherein generating the new computer software system from the business process using a predefined solution library includes  
25           generating business objects, generating business hierarchies, generating state models, and generating user interfaces.
5. The method according to claim 1 wherein generating the integration code from the new computer software systems into the third party software systems using predefined integration software components includes generating the mapping of data

models/data structures of the new computer software system and the third party computer software system using the business objects designed in the business process

6. A computerized method, comprising:  
providing mutually exclusive and completely exhaustive (MECE) software  
5 components for generating computer software systems for all business application across the industries, and

providing software components for generating integration code into pre-existing software systems for commonly used pre-existing packaged software systems

7. The method according to claim 6, wherein the mutually exclusive and  
10 completely exhaustive software components for generating computer software systems for all business application across the industries includes the following software components: business process templates, industry templates, and integration templates.

8. The method according to claim 7, wherein said business process  
templates includes: business objects, object attributes, relationships between objects,  
15 configurable behavior of the objects defined by the business object state models, display panels to visualize the business objects, and business patterns that provide the software functions of the template.

9. The method according to claim 7, wherein said industry templates  
includes: business objects, object attributes, relationships between objects, configurable  
20 behavior of the objects defined by the business object state models, display panels to visualize the business objects, and business patterns that provide the software functions of the template.

10. The method according to claim 7, wherein said industry templates  
includes: industry specific business objects, industry specific object attributes, industry  
25 specific relationships between objects, configurable behavior of the industry specific objects defined by the industry specific business object state models, industry specific display panels to visualize the business objects, and industry specific business patterns that provide the software functions of the template.

11. The method according to claim 7, wherein said integration templates  
30 includes: business objects, object attributes, relationships between objects, configurable

behavior of the objects defined by the business object state models, display panels to visualize the business objects, and business patterns that provide the software functions of the template.

12. The method according to claim 7, wherein said integration templates  
5 includes: integration derived business objects, integration derived object attributes, integration derived relationships between objects, integration derived configurable behavior of the objects defined by the integration derived business object state models, integration derived display panels to visualize the business objects, and integration derived business patterns that provide the software functions of the template.

10 13. An apparatus that allows a user to automatically generate a software computer system tailored to a particular business model, without programming by the user, comprising:

a software portion configured to facilitate the modeling of one or more  
business processes, business objects and business rules of the user to yield a business model  
15 having one or more business processes;

a software portion configured to optimize the business model;

a software portion configured to, if a business process does not correspond to a  
pre-existing computer software system, generate a new computer software system for the  
business process using a predefined software component library; and

20 a software portion configured to, if a business process corresponds to pre-existing computer software system, then generate integration code from new computer software systems into the pre-existing software systems using predefined integration software components.

25

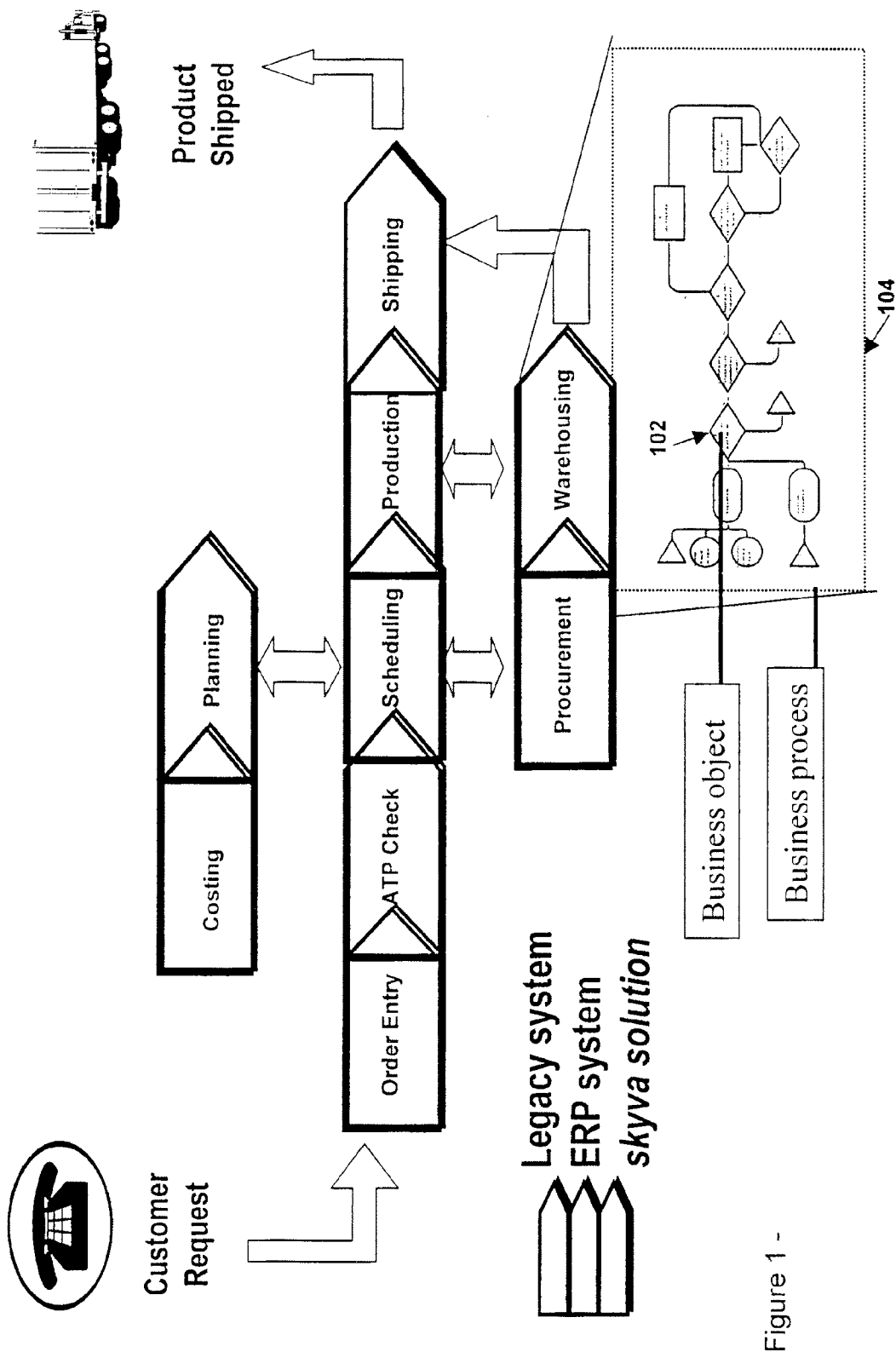


Figure 1 -

1/23

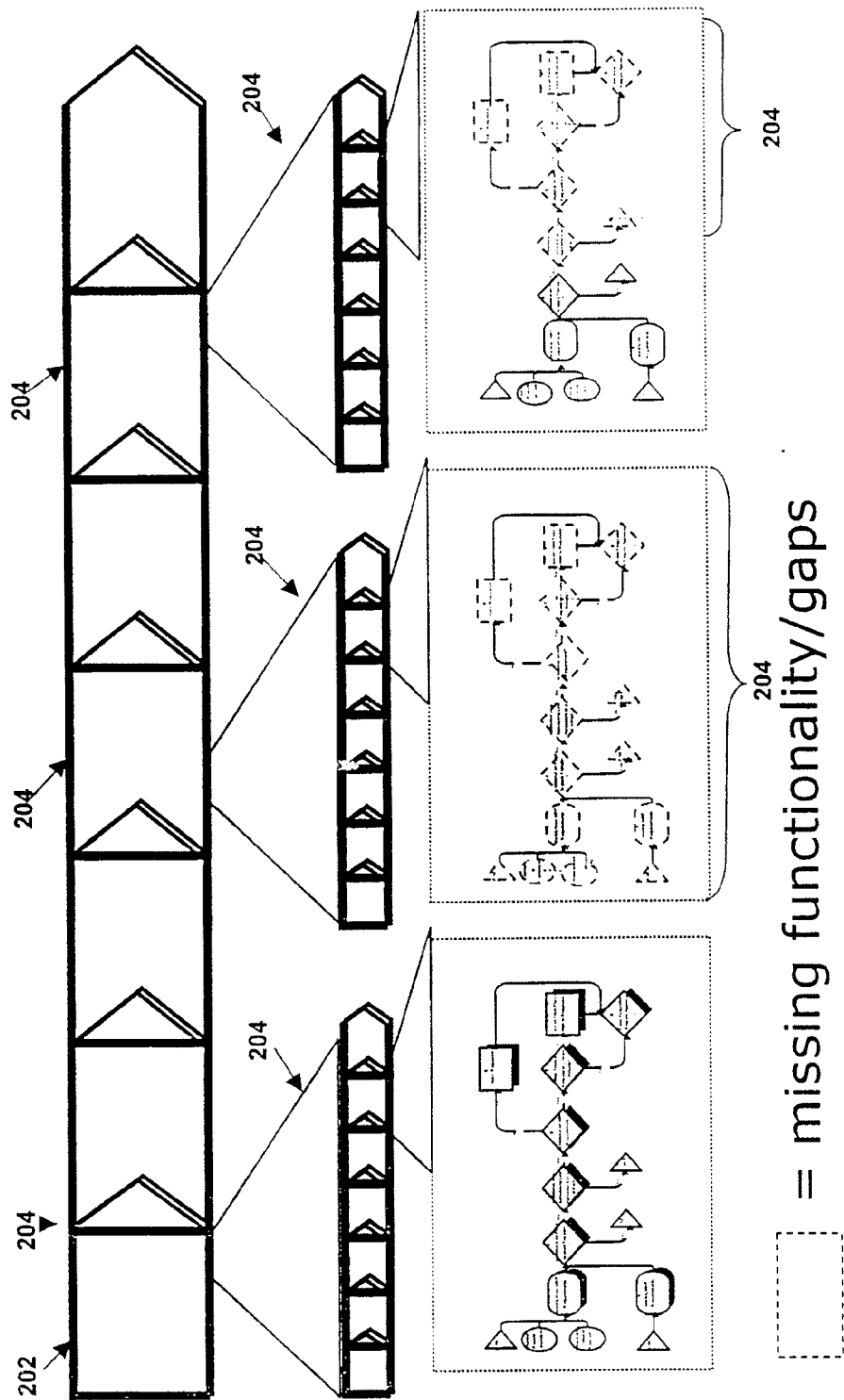


Figure 2 -

2/23

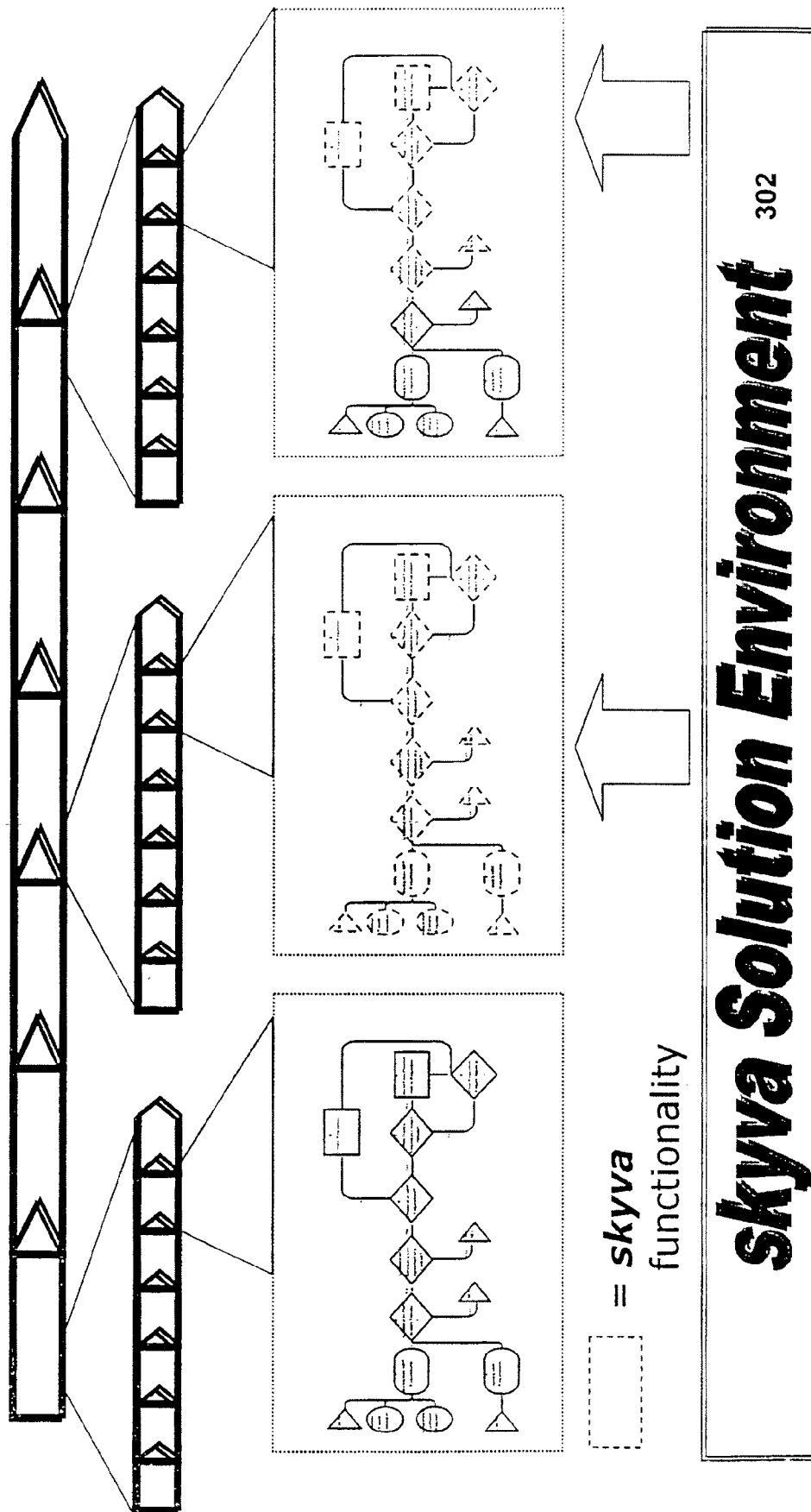


Figure 3 -

3/23

SUBSTITUTE SHEET (RULE 26)



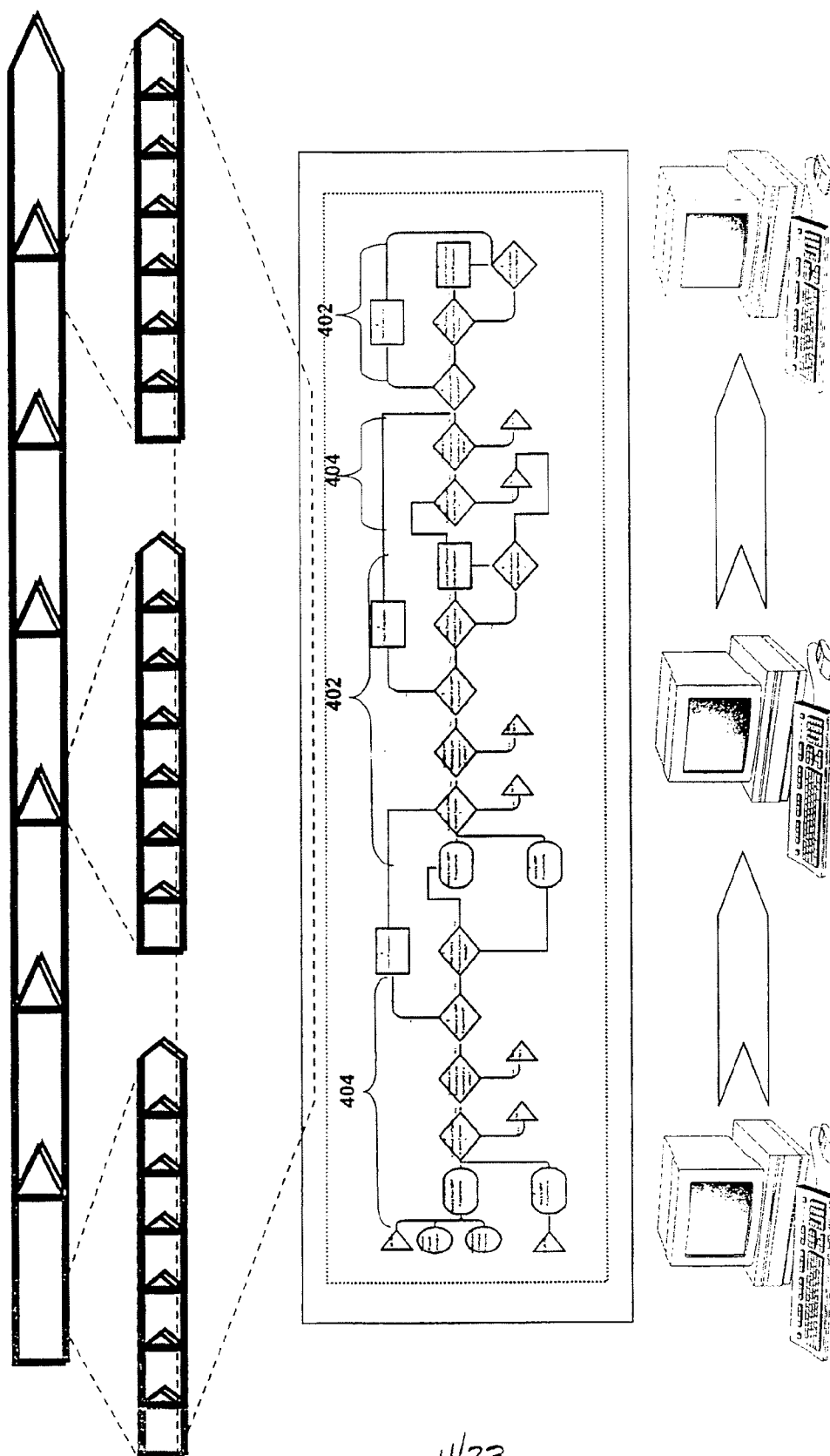


Figure 4 -

4/23

SUBSTITUTE SHEET (RULE 26)

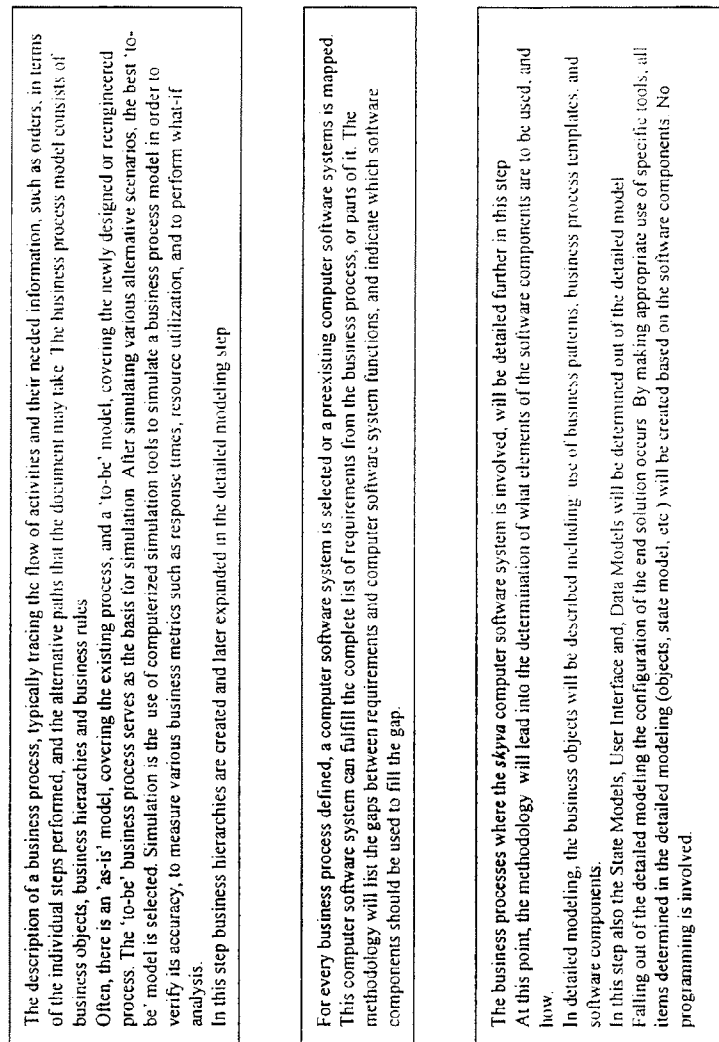


Figure 5 -

5/23

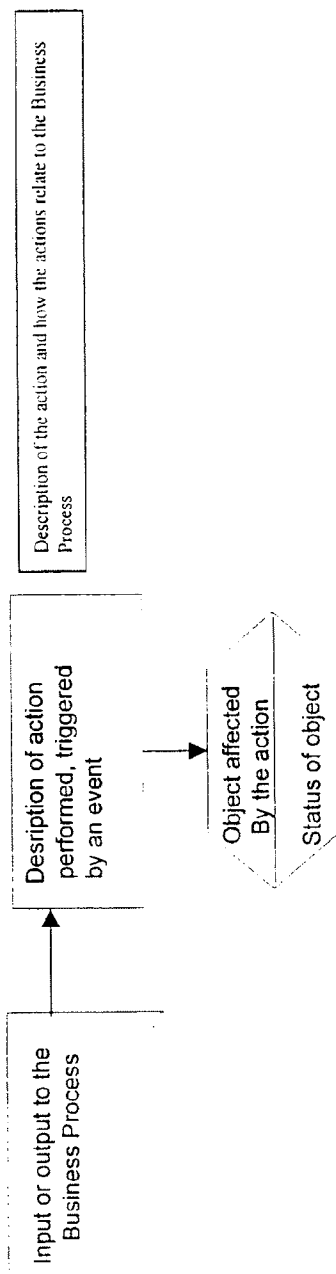


Figure 6 -

6/23

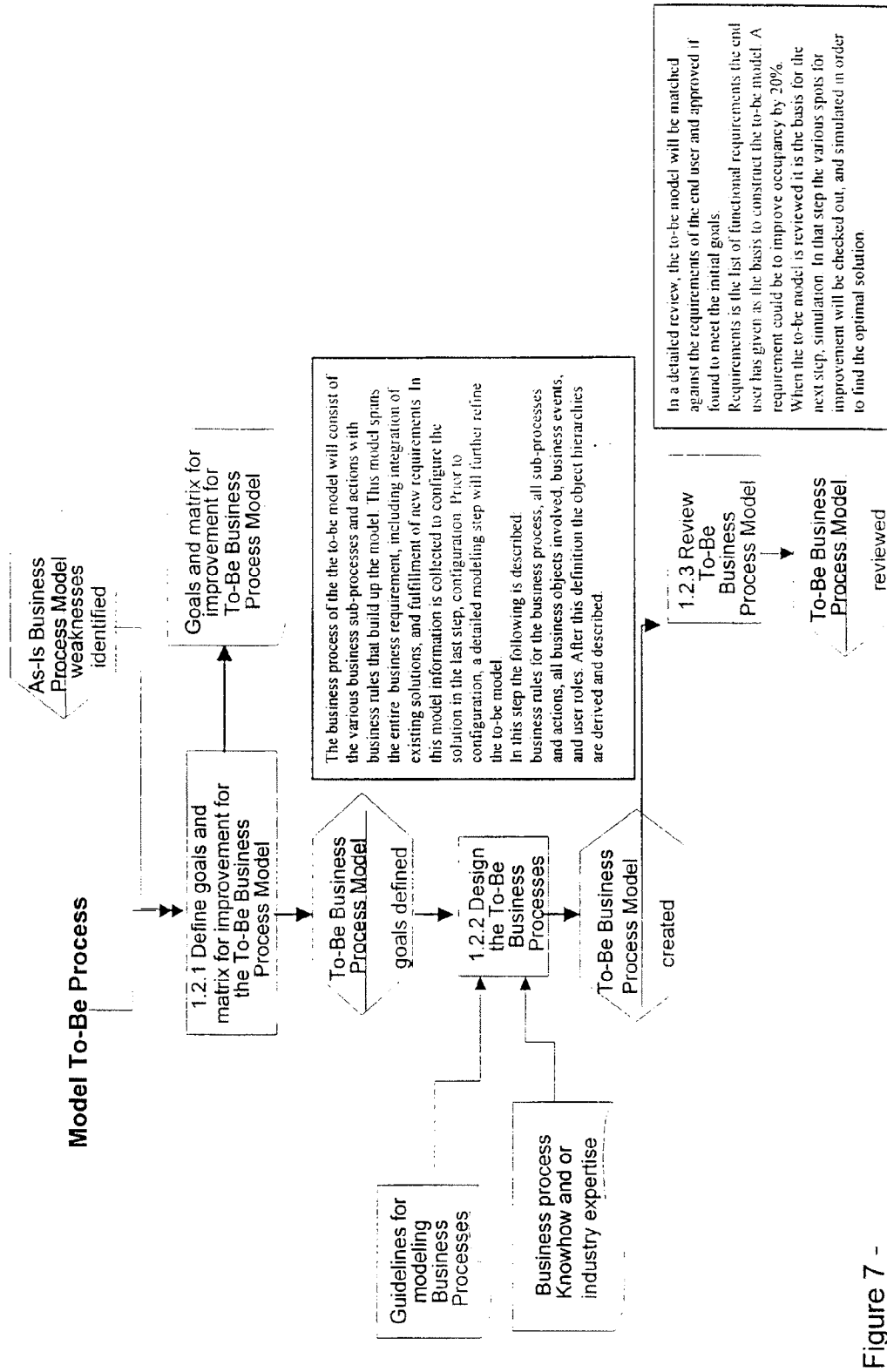


Figure 7 -

7/23

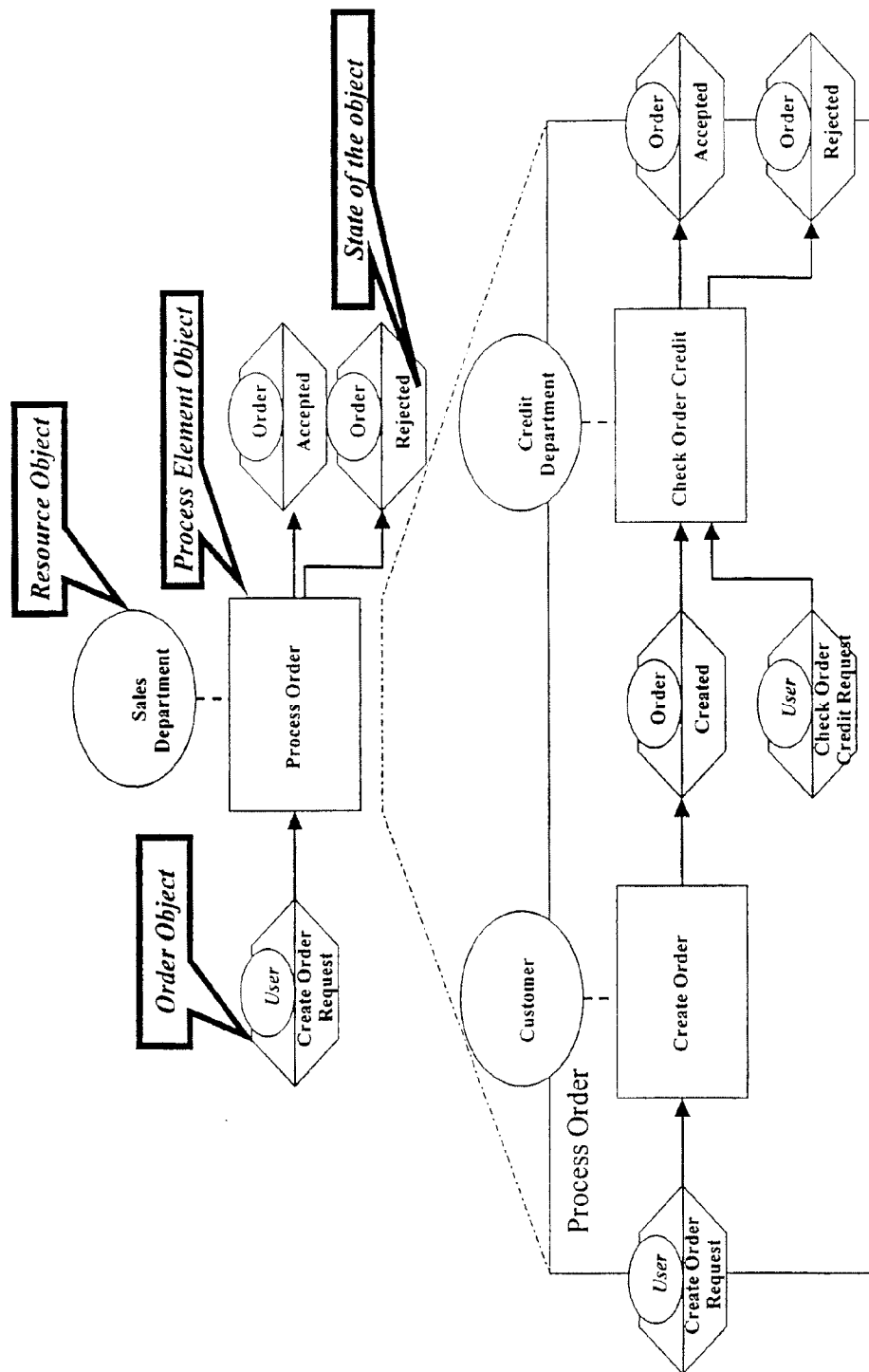


Figure 8 -

8/23

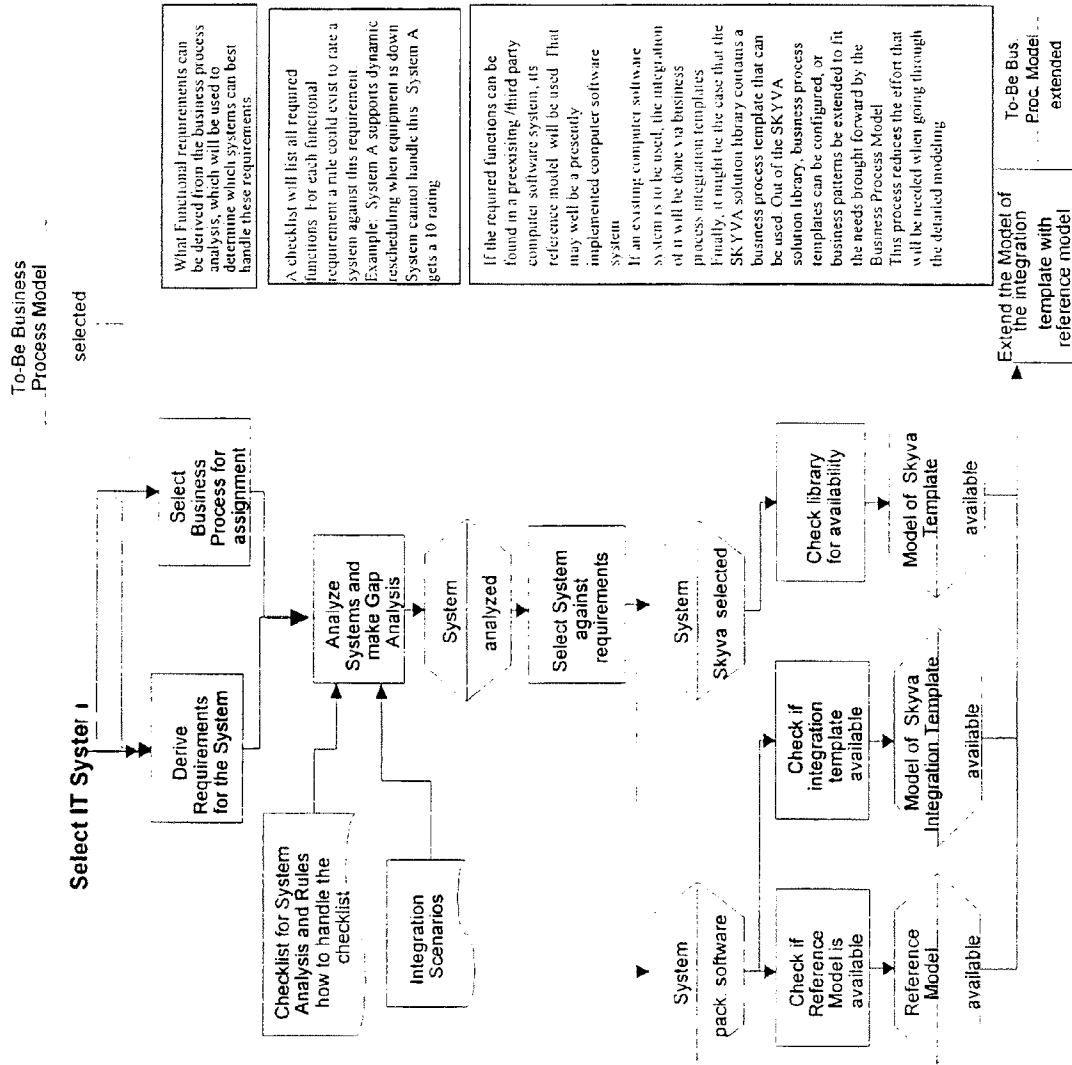


Figure 9 -

9/23

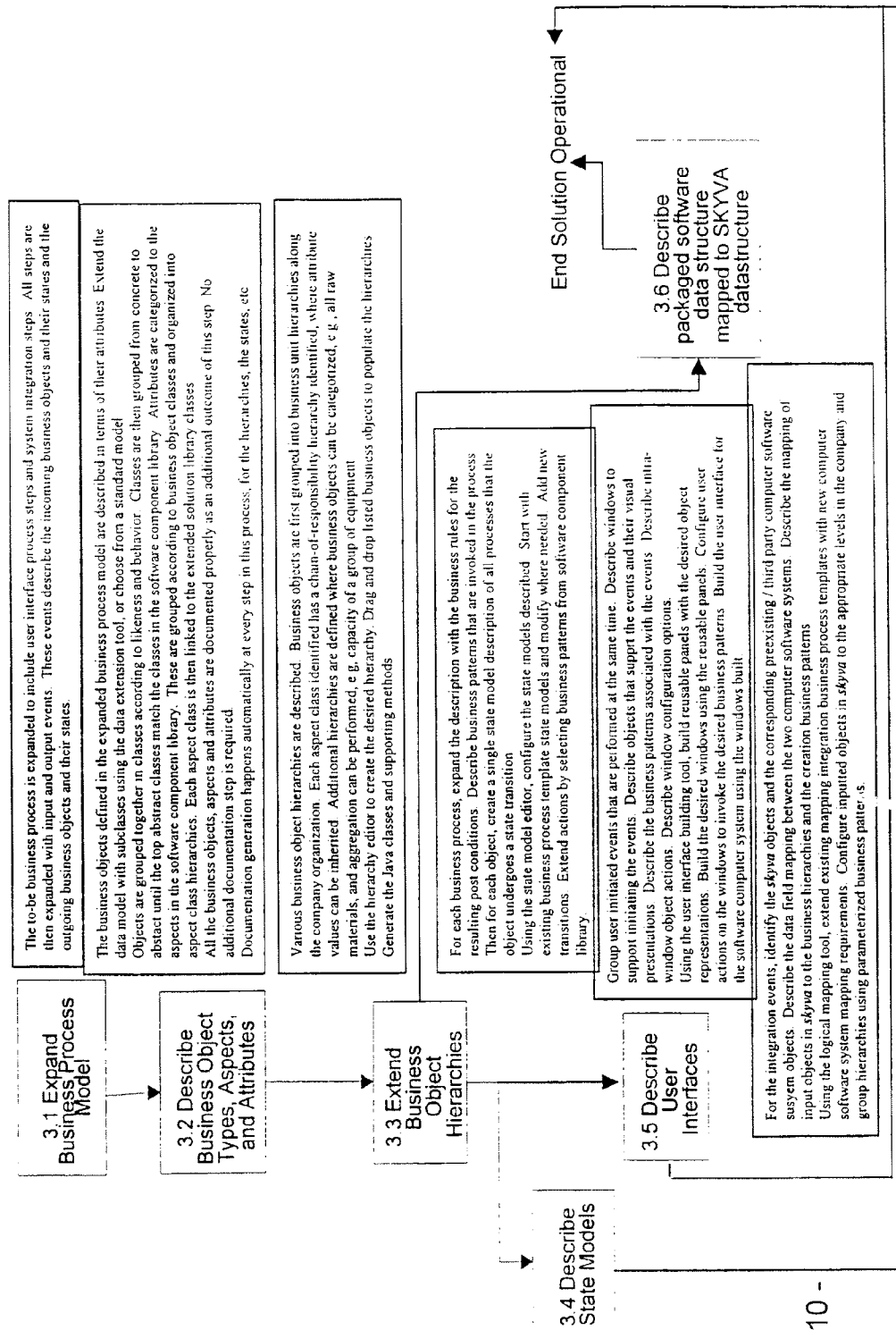


Figure 10 -

10/23

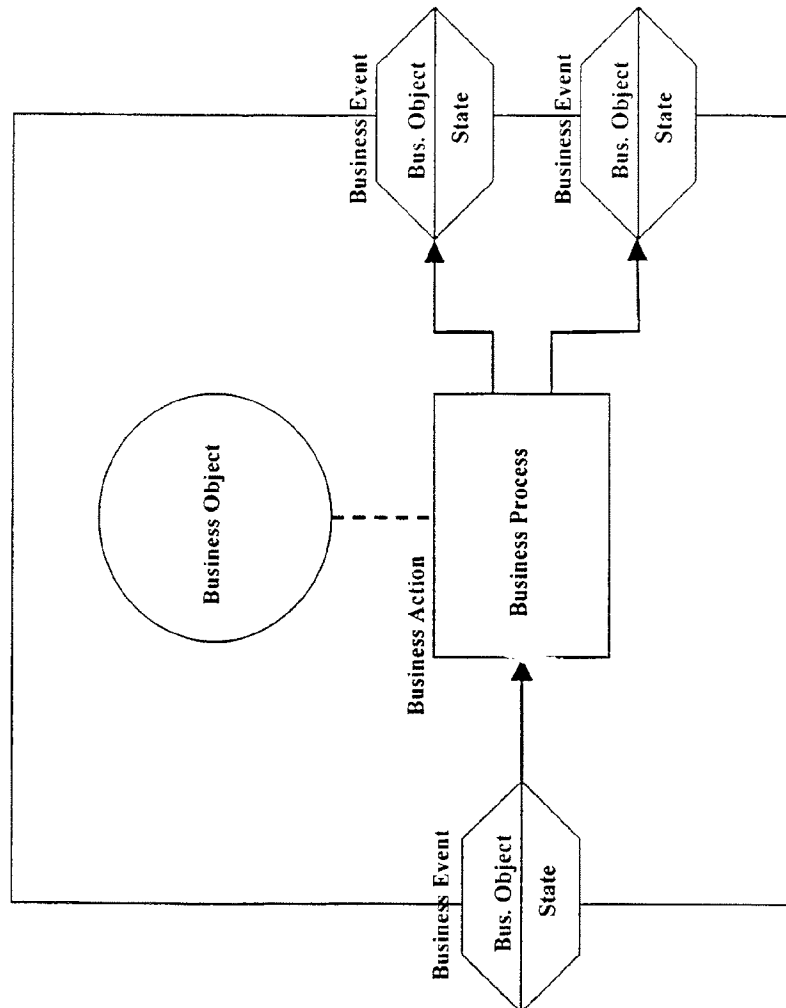


Figure 11

11/23



<b>Attribute</b>	<b>Format</b>	<b>Display Names</b>	<b>Description</b>
<b>Id</b>	String	<b>Short:</b> ID <b>Long:</b> ID	Unique identifier (Automatically set by the system)
<b>Name</b>	String	<b>Short:</b> {Short Descriptor up to 10 Char} <b>Long:</b> {Long Descriptor up to 15}	Name of the facility
<b>Deleted</b>	Boolean	<b>Short:</b> Deleted <b>Long:</b> Deleted	Deleted flag

Figure 12

12/23

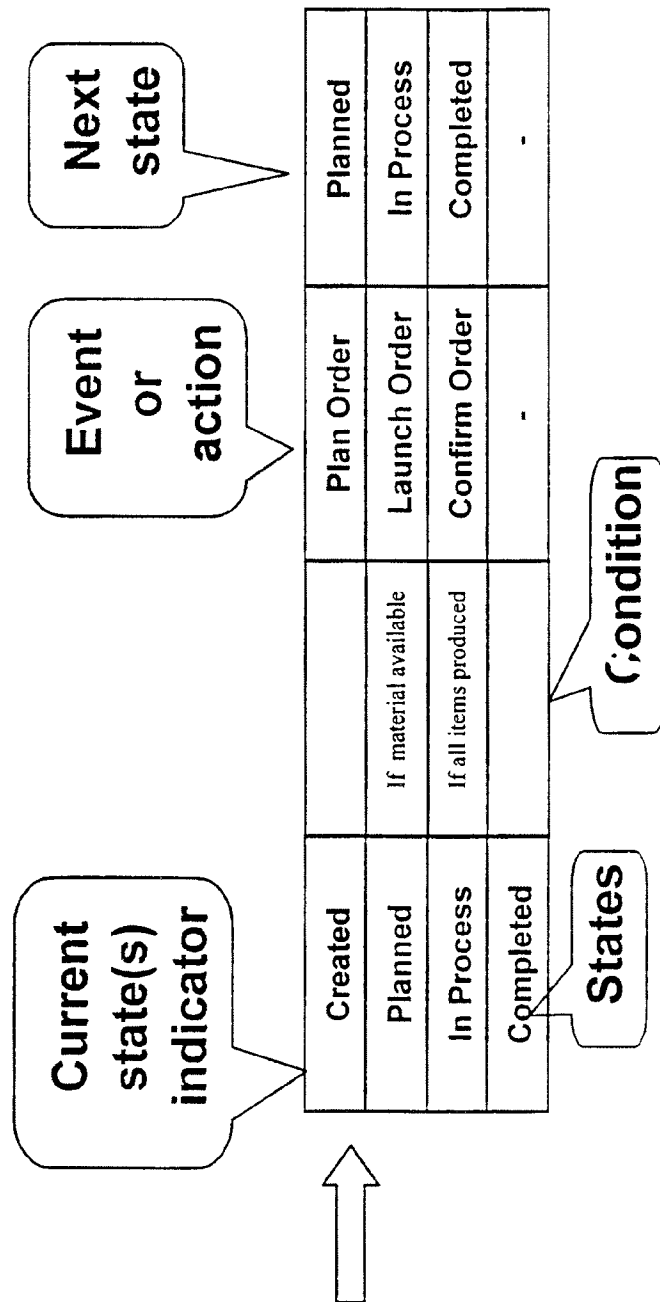


Figure 13

13/23

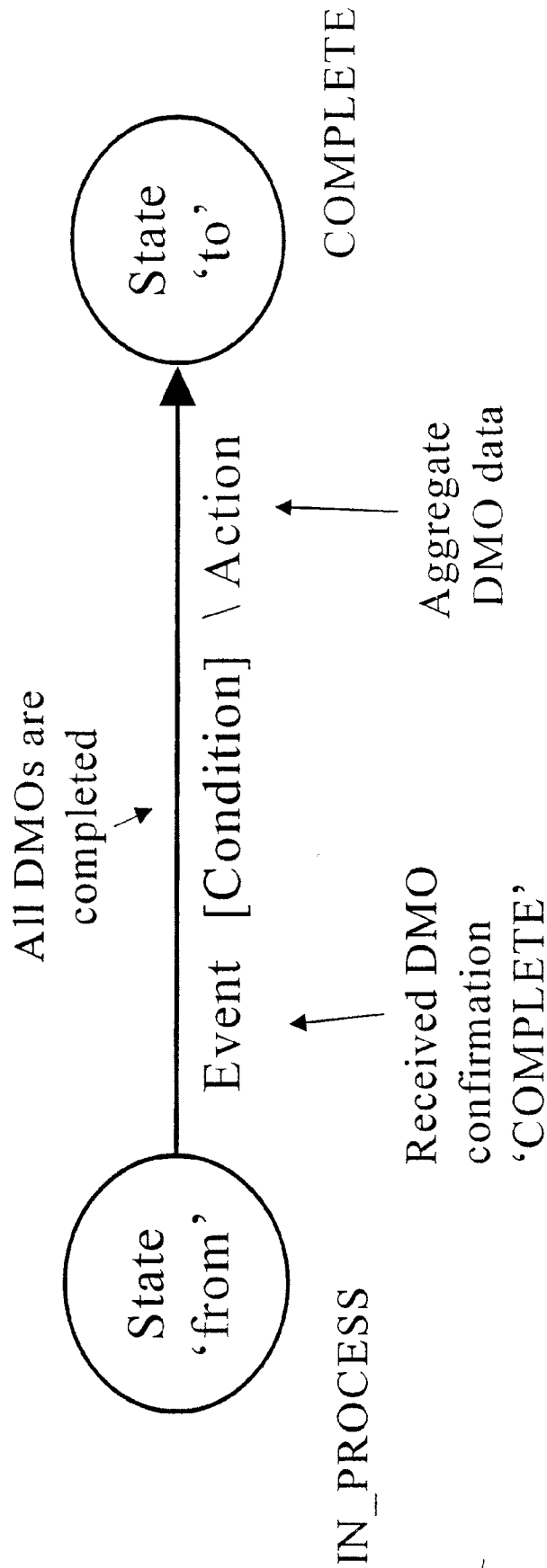


Figure 14

14/23

**ProductionOrder** [X]

### Production Orders

**Proposed Production Order**

Month	Material	Qty. Req.	Qty. Targ.
19980601	PUG	49500.000	

**Predicted Production Order**

Month	Material	Inventory	Quantity	Unit
5/1/98	ARL	217000	0	TO
5/1/98	AMN	93000	0	TO
5/1/98	VGO	92907	0	TO
5/1/98	PUG	48670	0	TO
5/1/98	MTBE	899	899	TO
5/1/98	LVN	16151	4929	TO
5/1/98	CCN	39959	20274	TO
5/1/98	GFK	62000	0	TO
5/1/98	REF	22568	22568	TO
5/1/98	ARES	178653	0	TO
5/1/98	HVN	29698	0	TO

Figure 15

15/23

SUBSTITUTE SHEET (RULE 26)

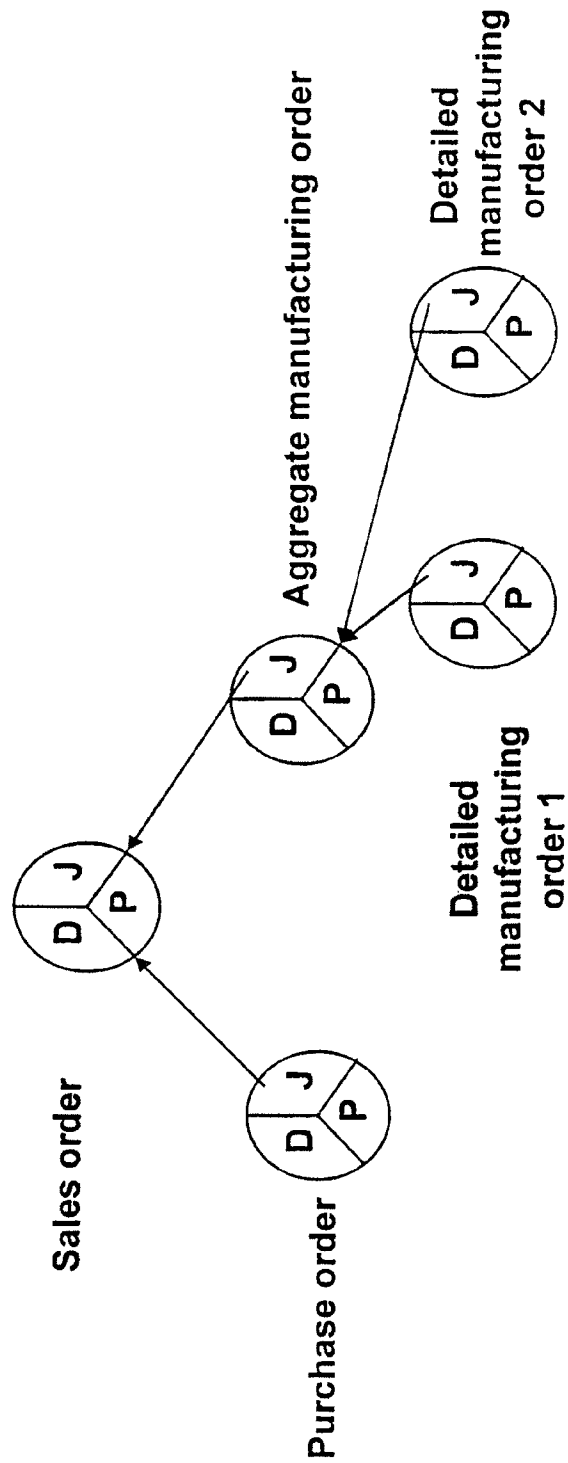


Figure 16

16/23

Attribute	Format	Display Names	Description
Id	Long	Short: ID Long: ID	Unique identifier
Name	String	Short: Man Order Long: Manufac Order	Name of the manufacturing order
Deleted	Boolean	Short: Deleted Long: Deleted	Deleted flag
Justification Aspect	N		
Justification Type	String	Short: Just.Type Long: JustificationType	Justification Type: Describe the type of the justification. It defines, if this justification is an Aggregation (Vertical Justification) or an Supply Requestor (horizontal justification) Possible Values • Aggregation • Supply Requestor
ReqOrder	Reference		Order whose requests created this request
ReqSupply	Reference		Primary supply, which is to be produced
ReqOrderQty	Decimal	Short: Req Qty Long: Requested Qty	Requested order quantity IDOC E2PLAFL/GSMNG
ReqOrderQtyUOM	Reference		UOM of the requested order quantity IDOC E2PLAFL-MEINS
ReqFacility	Reference		The requested primary facility of the requesting order
ReqDueDate	Time	Short: End Time Long: End Time	Requested end date of the order IDOC Customer request

Figure 17

17/23

SUBSTITUTE SHEET (RULE 26)

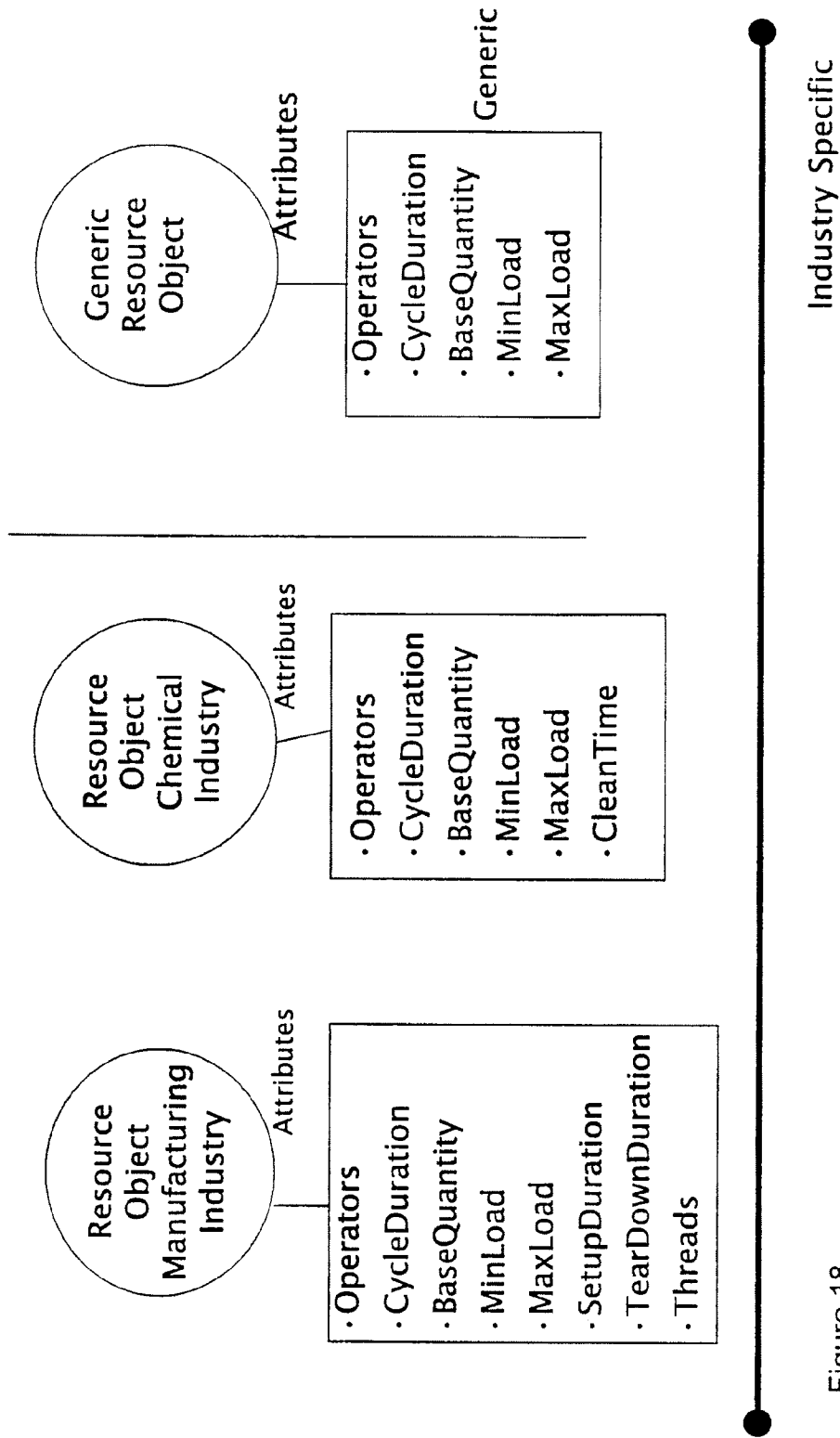


Figure 18

18/23

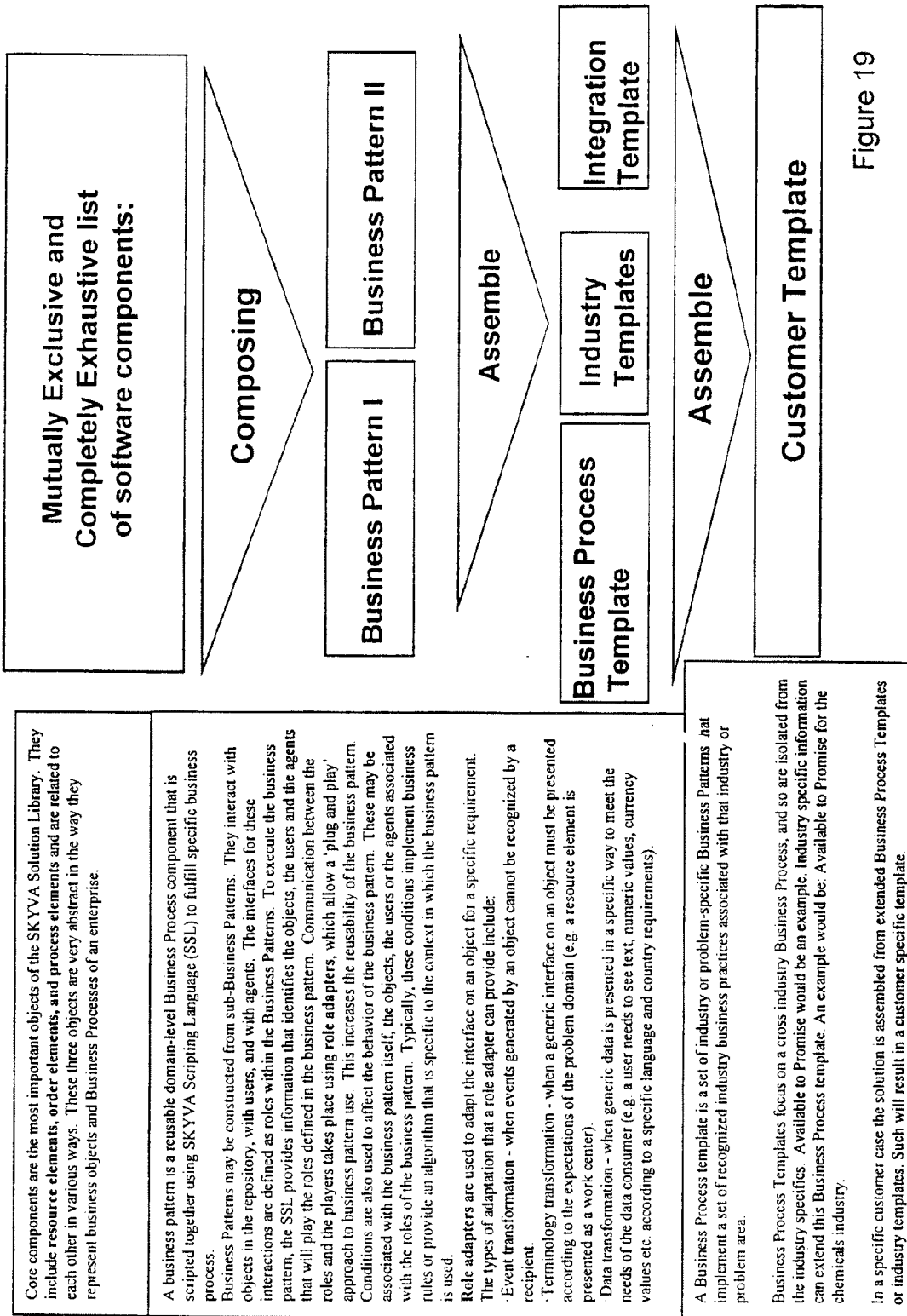
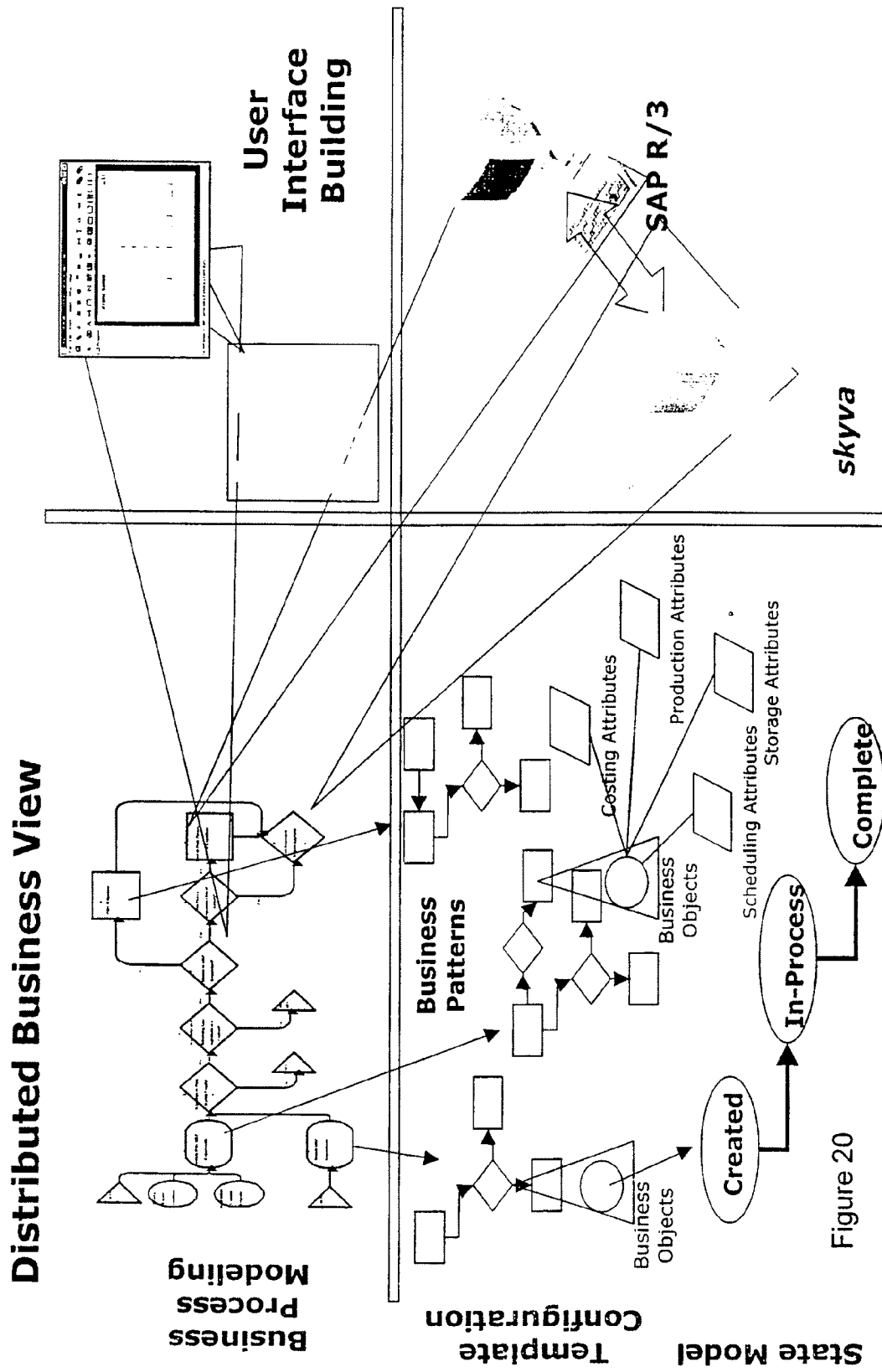


Figure 19

19/23





20/23

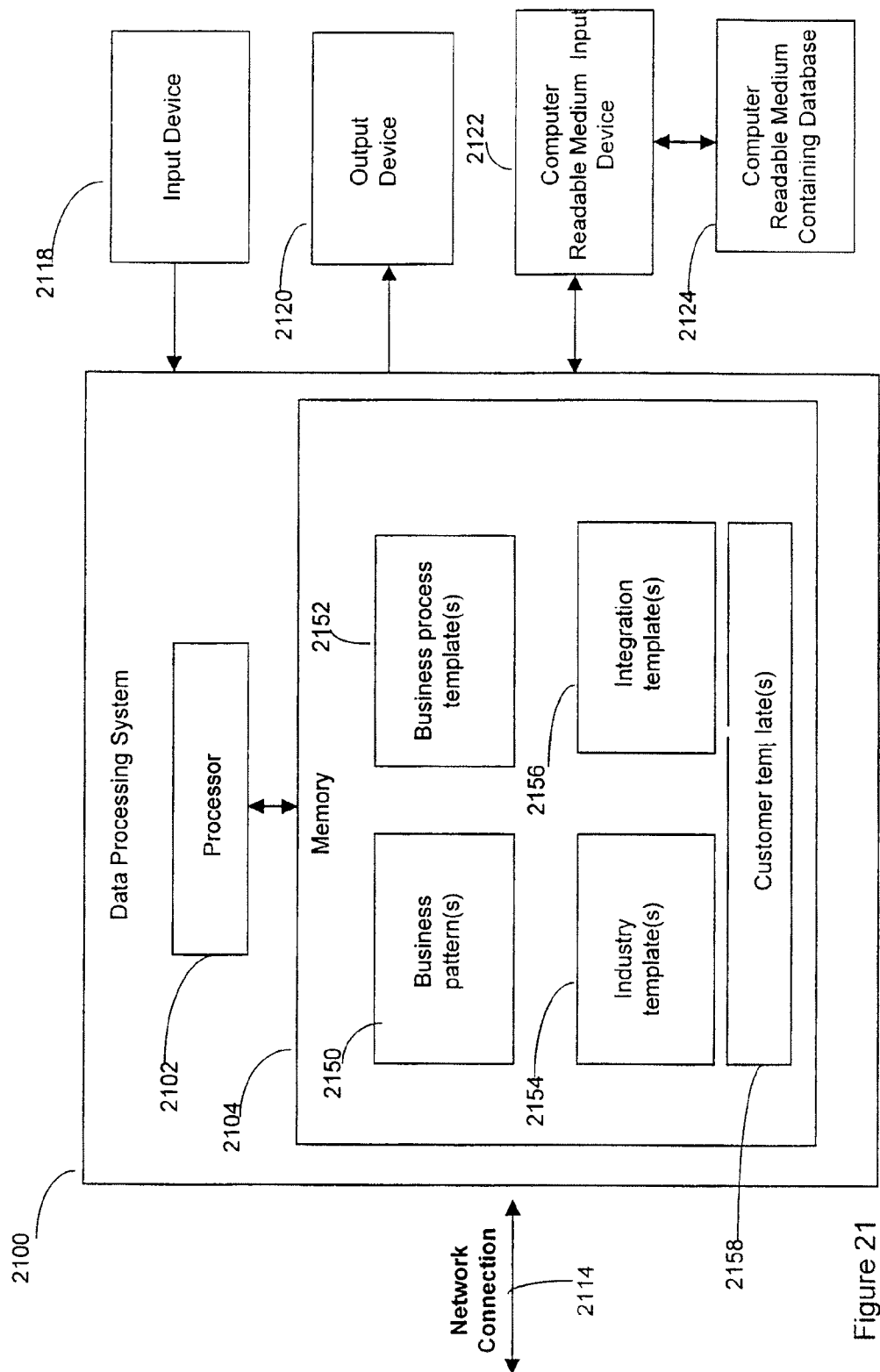


Figure 21

21/23

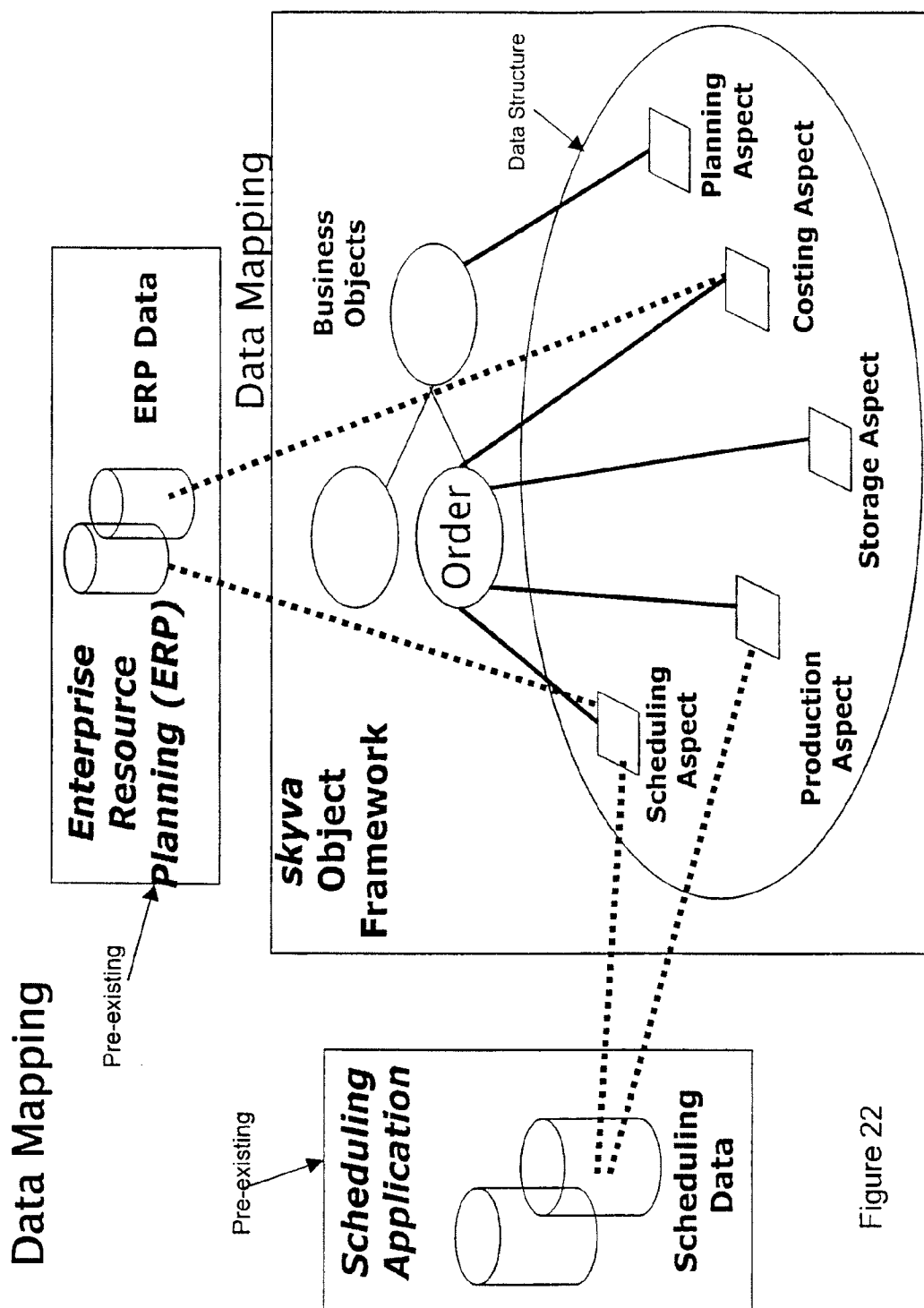


Figure 22

22/23

# Business Object Designer

**Skyva Business Object Designer**  
File Edit Tools

**Resource Elements** **Order Elements** **Process Elements** **Aspects**

**Module View**

- ☐ Discrete Mfg.
  - ☐ Batch
  - ☐ BusinessUnit
  - ☐ Facility
  - ☐ ManufacturingResource
    - ☐ Material
    - ☐ ObjectTypes
    - ☐ Post
    - ☐ PrintProfile
    - ☐ StorageUnit
    - ☐ Supply
    - ☐ Tool
    - ☐ User
- ☐ Weaverville Plant
  - ☐ CdAtropackKing
  - ☐ CdReplication
  - ☐ DwlReplication
  - ☐ Electrophilating
  - ☐ Kitting
  - ☐ LaserBeamReconderAnd
  - ☐ McDuplication
  - ☐ McLoading
  - ☐ McPrintPack

**Business Object View**

- ☐ ResourceElement
  - ☐ BusinessUnit
    - ☐ Facility
      - ☐ ManufacturingResource
        - ☐ McLoading
          - ☐ AcquisitionAspect
          - ☐ BatchAspect
          - ☐ CapacityAvailabilityAspect
          - ☐ CapacityUnitAspect
          - ☐ IntegrationsAspect
          - ☐ LocationAspect
          - ☐ OrganizationalUnitAspect
          - ☐ PerformanceAspect
          - ☐ ProductionAspect
          - ☐ ResourceCapabilityAspect
          - ☐ SAPAspect
          - ☐ SizesAspect
          - ☐ StockingAspect
          - ☐ ToolAspect
          - ☐ VariablesAspect

This aspect provides information needed to calculate the capacity load and the execution time during planning and scheduling

Name	Type
Operators	Text
Threads	Integer
SetupDuration	Duration
TotalDownDuration	Duration
CycleDuration	Text
BaseQuantity	AmountOfTime
MinLoad	UnitOfMeasure
MaxLoad	QuantityUnit
	MeasurementValue
	Currency
	Formula
	Reference

Figure 23

23/23

SUBSTITUTE SHEET (RULE 26)

## INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 99/27883

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, IBM-TDB

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 722 140 A (INST OF SOFTWARE SCIENT CONST ;JR EAST JAPAN INFORMATION SYST (JP)) 17 July 1996 (1996-07-17) page 2, line 1 -page 4, line 16 ----	1-13
A	EP 0 895 156 A (KNOWLEDGE MODELING INST INC) 3 February 1999 (1999-02-03) column 1, line 1 -column 2, line 52 ----	1-13
A	KIEM-PHONG VO: "IFS: A TOOL TO BUILD APPLICATION SYSTEMS" IEEE SOFTWARE,US,IEEE COMPUTER SOCIETY. LOS ALAMITOS, vol. 7, no. 4, 1 July 1990 (1990-07-01), pages 29-36, XP000137895 ISSN: 0740-7459 the whole document -----	1-13



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

## \* Special categories of cited documents :

- \*A\* document defining the general state of the art which is not considered to be of particular relevance
- \*E\* earlier document but published on or after the international filing date
- \*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- \*O\* document referring to an oral disclosure, use, exhibition or other means
- \*P\* document published prior to the international filing date but later than the priority date claimed

\*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

\*X\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

\*Y\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

\*G\* document member of the same patent family

Date of the actual completion of the international search

22 August 2000

Date of mailing of the international search report

29/08/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Brandt, J

# INTERNATIONAL SEARCH REPORT

information on patent family members

International Application No

PCT/US 99/27883

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0722140 A	17-07-1996	US 5809304 A WO 9602033 A	15-09-1998 25-01-1996
EP 0895156 A	03-02-1999	JP 11053181 A JP 11053182 A AU 701351 A	26-02-1999 26-02-1999 28-01-1999